

# Generación de conocimiento de una web, sistemas expertos

---

FACULTAD DE INFORMÁTICA DE BARCELONA (FIB)

UNIVERSIDAD POLITÉCNICA DE CATALUÑA (UPC)

**Autor: Víctor Mateos Matilla**

**Ponente: Carme Martin Escofet**

**Directora: Patricia Picanyol Mercadal**

**Especialidad Ingeniería del Software**

# Índice

<b>Índice .....</b>	<b>2</b>
<b>1 Resumen .....</b>	<b>8</b>
1.1 Resum .....	8
1.2 Abstract.....	8
<b>2 Contexto .....</b>	<b>9</b>
2.1 Introducción.....	9
2.2 Actores implicados.....	9
2.3 Motivación personal .....	9
2.4 Objetivos del proyecto.....	10
<b>3 Estado del arte .....</b>	<b>11</b>
3.1 Sistemas existentes.....	11
3.1.1 Hojas de cálculo.....	11
3.1.2 Analíticas web .....	11
3.1.3 Sistemas expertos propios.....	11
3.1.4 Microsoft Azure ML .....	12
3.1.5 Google Prediction Api.....	12
3.1.6 Big ML.....	12
3.1.7 PredicSis.....	13
3.1.8 IBM Watson Analytics .....	13
3.2 Conclusiones sistemas actuales .....	13
<b>4 Formulación del problema .....</b>	<b>14</b>
<b>5 Alcance.....</b>	<b>15</b>
5.1 Herramientas utilizadas .....	16
5.1.1 Liferay 6.2 EE .....	16
5.1.2 Bases de datos de Amazon, Amazon RDS .....	17
5.1.3 Sistema de almacenamiento, Amazon S3 .....	17
5.1.4 Sistemas expertos de Amazon, Amazon Machine Learning.....	17
5.2 Justificación de las tecnologías .....	18
5.3 Obstáculos y planes de contingencia.....	18

<b>6 Metodología y rigor .....</b>	<b>19</b>
6.1 Metodología ágil .....	19
6.2 Herramientas de seguimiento .....	19
6.3 Método de validación .....	19
<b>7 Planificación temporal inicial .....</b>	<b>20</b>
7.1 Calendario .....	20
<b>8 Iteraciones de desarrollo .....</b>	<b>21</b>
8.1 Sprint Inception: Investigación, planificación y puesta en marcha .....	21
8.2 Sprint 1: Implementación REST API y documentación .....	21
8.3 Sprint 2: Configuración sistema experto y documentación .....	21
8.4 Sprint 3: Implementación web - REST API - sistema experto y documentación ..	22
8.5 Sprint 4: Documentación final .....	22
<b>9 Tabla de costes temporales.....</b>	<b>23</b>
<b>10 Diagrama de Gantt.....</b>	<b>24</b>
10.1 Parte Diagrama: Sprint Inception .....	24
10.2 Parte Diagrama: Sprint 1, 2, 3, 4.....	25
<b>11 Valoración de alternativas y plan de acción .....</b>	<b>26</b>
<b>12 Análisis económico inicial .....</b>	<b>27</b>
12.1 Costes directos por actividad.....	27
12.1.1 Amortización de la maquinaria .....	28
12.1.2 Amortización del software .....	29
12.2 Costes indirectos.....	29
12.2.1 Conexión de datos .....	29
12.2.2 Transporte .....	29
12.2.3 Impresión en papel .....	29
12.3 Contingencia .....	30
12.4 Imprevistos .....	30
12.4.1 Retrasos.....	30
12.4.2 Avería de la maquinaria .....	30
12.5 Presupuesto .....	31
12.6 Control de gestión.....	31
<b>13 Sostenibilidad .....</b>	<b>32</b>

13.1 Ambiental.....	33
13.2 Económica.....	33
13.3 Social .....	34
<b>14 Análisis de requisitos .....</b>	<b>35</b>
14.1 Requisitos funcionales .....	35
14.1.1 Casos de Uso WEB.....	35
14.1.1.1 Diagrama.....	35
14.1.1.2 Seleccionar Vehículo.....	36
14.1.1.3 Seleccionar Partes implicadas .....	36
14.1.1.4 Seleccionar Información .....	36
14.1.1.5 Seleccionar Coche.....	36
14.1.1.6 Seleccionar Moto.....	37
14.1.1.7 Seleccionar UPC.....	37
14.1.1.8 Seleccionar Everis .....	37
14.1.1.9 Seleccionar Datos .....	37
14.1.2 Casos de Uso REST API .....	38
14.1.2.1 Diagrama.....	38
14.1.2.2 Mostrar información .....	38
14.1.2.3 Guardar información .....	38
14.1.2.4 Distribuir información .....	39
14.1.2.5 Almacenar datos para analizarlos .....	39
14.1.3 Casos de Uso Sistemas Amazon .....	40
14.1.3.1 Diagrama.....	40
14.1.3.2 Iniciar sesión.....	40
14.1.3.3 Cerrar sesión.....	41
14.1.3.4 Crear BD .....	41
14.1.3.5 Crear almacenamiento datos.....	41
14.1.3.6 Crear modelo de aprendizaje .....	42
14.2 Requisitos no funcionales .....	42
14.2.1 Fácil de utilizar para cualquier tipo de usuario .....	42
14.2.2 Reutilizable y personalizable para cualquier empresa e organización .....	42
14.2.3 Garantiza la seguridad de la información .....	42

14.2.4 Buena apariencia.....	43
14.2.5 Alta disponibilidad y escalabilidad .....	43
14.2.6 Rápido .....	43
14.2.7 Usable.....	43
14.2.8 Iconos convencionales .....	43
14.2.9 Actualizaciones.....	43
14.2.10 Soporte constante .....	43
14.3 Diagramas .....	44
14.3.1 Diagrama Modelo conceptual de datos.....	44
14.3.2 Descripción Modelo conceptual de datos .....	44
14.3.3 Diagrama UML del portal web.....	45
14.3.4 Descripción UML del portal web .....	45
14.3.5 Diagrama secuencia distribuir información .....	46
14.3.6 Diagrama secuencia mostrar información.....	48
<b>15 Diseño final .....</b>	<b>49</b>
15.1 Diseño portal web.....	49
15.1.1 Página principal.....	49
15.1.2 Página secundaria Coche .....	50
15.1.3 Página secundaria Moto .....	50
15.1.4 Página secundaria Partes implicadas.....	51
15.1.5 Página secundaria UPC .....	51
15.1.6 Página secundaria Everis.....	52
15.1.7 Página secundaria Información .....	52
15.1.8 Página secundaria Datos.....	53
15.2 Diseño Rest API .....	53
15.2.1 Mostrar información .....	53
15.2.2 Almacenar datos para analizarlos .....	54
<b>16 Arquitectura de Liferay .....</b>	<b>55</b>
16.1 Instancias .....	55
16.1.1 Sites .....	55
16.1.2 Usuarios.....	56
16.1.3 Roles .....	56

16.1.4 Jerarquía .....	57
16.2 Arquitectura .....	57
16.3 Portlet .....	59
<b>17 Implementación .....</b>	<b>60</b>
17.1 Tecnologías .....	60
17.2 Obtener información interacción web .....	61
17.3 Rest Api .....	62
17.4 Análisis e interpretación de datos .....	65
17.4.1 Requisitos .....	65
17.4.2 Amazon RDS .....	65
17.4.3 Amazon IAM .....	67
17.4.4 Amazon S3 .....	67
17.4.5 Amazon Machine Learning .....	68
17.4.6 Toad MySQL .....	73
17.4.7 MySQL Workbench .....	73
<b>18 Planificación, análisis de costes y metodologías .....</b>	<b>74</b>
18.1 Modificaciones realizadas en la planificación .....	74
18.1.1 Sprint Inception: Investigación, planificación y puesta en marcha .....	74
18.1.1.1 Sprint inception final .....	75
18.1.2 Sprint 1: Implementación REST API y documentación .....	75
18.1.2.1 Sprint 1 final .....	76
18.1.3 Sprint 2: Configuración sistema experto y documentación .....	76
18.1.3.1 Sprint 2 final .....	76
18.1.4 Sprint 3: Implementación web - REST API - sistema experto y documentación .....	77
18.1.4.1 Sprint 3 final .....	77
18.1.5 Sprint 4: Documentación final .....	77
18.1.5.1 Sprint 4 final .....	78
18.1.6 Diagrama de Gantt final .....	79
18.1.6.1 Parte Diagrama de Gantt final: Sprint Inception .....	79
18.1.6.2 Parte Diagrama de Gantt final: Sprint 1, 2, 3, 4 .....	80
18.2 Análisis económico final .....	81

18.2.1 Costes directos por actividad .....	81
18.2.2 Costes indirectos.....	83
18.2.3 Presupuesto final.....	83
18.3 Metodología.....	84
<b>19 Integración de conocimientos .....</b>	<b>85</b>
19.1 ASW (Aplicaciones y Servicios Web).....	85
19.2 DBD y BD (Diseño de Base de Datos y Base de Datos) .....	85
19.3 PES (Proyecto de Ingeniería del Software) .....	85
19.4 PROP (Proyectos de Programación).....	85
19.5 APC y CSI (Arquitectura del PC y Conceptos de sistemas de información) .....	85
<b>20 Listado de competencias técnicas del proyecto .....</b>	<b>86</b>
<b>21 Leyes y Regulaciones .....</b>	<b>87</b>
<b>22 Mejoras .....</b>	<b>88</b>
<b>23 Situación actual y futuro .....</b>	<b>89</b>
<b>24 Conclusiones.....</b>	<b>90</b>
<b>25 Referencias.....</b>	<b>91</b>
<b>26 Glosario .....</b>	<b>92</b>

# 1 Resumen

Tener datos y saber hacer un correcto análisis e interpretación de estos es una de las claves en el éxito empresarial. Las empresas son conscientes de ello e invierten grandes recursos, ya sean de personal o material, pero cada vez el volumen de datos a tratar es mayor y se necesitan sistemas más eficientes.

El proyecto que se explica a continuación en detalle, soluciona esta problemática con un sistema capaz de obtener, estructurar, distribuir y analizar los datos de la interacción de un usuario por un portal web.

## 1.1 Resum

Tenir dades i saber fer un correcte anàlisi i interpretació d'aquestes és una de les claus en l'èxit empresarial. Les empreses són conscients d'això i inverteixen grans recursos, ja siguin de personal o material, però cada cop el volum de dades a tractar és més gran i es necessiten sistemes més eficients.

El projecte que s'explica a continuació en detall, soluciona aquesta problemàtica amb un sistema capaç d'obtenir, estructurar, distribuir i analitzar les dades de la interacció d'un usuari per un portal web.

## 1.2 Abstract

Collecting data and knowing how to make a proper analysis and interpretation of it is the key to the business success. Corporations are aware of this and invest big amount of resources, personal and material, but as the amount of data to analyse increases, more efficient systems are needed.

The following project, solves these problems with a system that is able to obtain, structure, distribute and analyse the data provided by the interaction of a user in a website.



## 2 Contexto

### 2.1 Introducción

Este proyecto se ha realizado como trabajo de final de grado de los estudios de Grado en Ingeniería Informática, en la especialidad de Ingeniería del Software, en la Facultad de Informática de Barcelona (FIB), de la Universidad Politécnica de Cataluña (UPC).

Este proyecto surge de la necesidad por parte de la empresa Everis<sup>1</sup> de ampliar y mejorar las funcionalidades que ofrecen a sus clientes. Concretamente la funcionalidad que necesitan ya que no disponen de un sistema para ello y que se ha desarrollado, es un sistema capaz de estructurar, distribuir e interpretar todos los datos que se quieran obtener de la interacción de un usuario con un portal web.

Para ello se ha creado una REST API (Representational State Transfer, Application Programming Interface) capaz de obtener, estructurar y distribuir estos datos, que posteriormente son analizados e interpretados por un sistema experto, que es un sistema software que simula el proceso de aprendizaje y que nos permitirá ser capaces de extraer conclusiones, aprender de los datos, y tomar decisiones y realizar acciones en función de los conocimientos obtenidos sobre esa gran cantidad de datos.

Por ser un proyecto de final de carrera y por temas de confidencialidad, el proyecto expuesto a continuación es un trabajo completamente propio, que se alinea en la medida de lo posible al proyecto real de Everis con sus clientes finales.

### 2.2 Actores implicados

Las partes interesadas e implicadas en el proyecto son aquellas que tienen un interés sobre el mismo. En este caso podemos identificarlos en los siguientes grupos:

- **Everis:** La empresa Everis tiene como objetivo desarrollar y mantener un sistema capaz de estructurar, distribuir e interpretar los datos obtenidos de sus portales web.
- **Cientes de Everis:** Los clientes de Everis serán la parte interesada que se beneficie directamente de lo que proporciona este sistema. Concretamente podrán extraer conclusiones y tomar decisiones a través de la interpretación de las datos recogidos de la interacción de los usuarios por sus portales webs.

### 2.3 Motivación personal

Han sido muchas las razones que me motivaron a realizar este Trabajo de Final de Carrera:

El motivo principal para la realización de este proyecto ha sido el interés que tengo en el mundo web, que ha sido la razón principal de cursar esta carrera de Ingeniería

---

<sup>1</sup> Consultora multinacional en la que realizo el proyecto de final de carrera

Informática en la UPC. Este proyecto ha sido una gran oportunidad para poder aplicar los conocimientos que he ido adquiriendo a lo largo de la carrera.

Otra motivación, ha sido la de tener que estudiar y aprender tecnologías nuevas muy utilizadas en el entorno empresarial como Liferay<sup>2</sup>, y también sobre sistemas expertos, a la par que creaba un prototipo de producto innovador que en el futuro podrá ser utilizado y mejorado.

## 2.4 Objetivos del proyecto

Los objetivos principales del proyecto son:

- ***Creación de una REST API***

Crear una REST API capaz de obtener, estructurar y distribuir información e datos de una web, teniendo una estructura de datos lo más reutilizable posible. El formato que se utilizara para estructurar los datos será JSON/XML, ya que es un formato que cualquier plataforma web es capaz de interpretar.

- ***Creación de una web***

Crear un portal web, el cual nos permitirá hacer la navegación de los usuarios por un portal web.

- ***Interpretación y análisis de datos en un sistema experto***

Mediante la REST API, distribuiremos e introduciremos los datos obtenidos de la manera más eficiente y estructurada posible en un sistema experto, de tal manera que este nos proporcione conclusiones lo más fiables posibles sobre ellos.

Otro objetivo que me gustaría mencionar debido a su importancia aunque ya viene implícito en el proyecto, es el de adquirir un conocimiento avanzado del uso e integración de distintas tecnologías, en especial Liferay 6.2.

---

<sup>2</sup> Portal de gestión de contenidos de código abierto escrito en Java.

## 3 Estado del arte

### 3.1 Sistemas existentes

Aunque es difícil encontrar un sistema completo similar al nuestro, ya que se trata de un sistema interno (privado) y que engloba diferentes fases. Hemos llevado a cabo un estudio de los sistemas que existen y se utilizan en la actualidad para estructurar, analizar e interpretar grandes cantidades de datos. A continuación explicaremos detalladamente algunos de los más relevantes.

#### 3.1.1 Hojas de cálculo

Uno de los sistemas más básicos, pero que en cambio está muy extendido en el mundo empresarial para estructurar y analizar datos, son las hojas de cálculo tipo Excel. Este sistema es muy engorroso, lento y costoso ya que necesitan que una o varias personas estén exclusivamente dedicadas a ello. Además el factor de riesgo o error al analizar estos es mayor que en otro tipo de sistemas. En cambio como factor favorable las empresas no necesitan tener desplegado un sistema específico para ello, ya que las hojas de cálculo están en la actualidad disponibles en cualquier pc, y además no necesitan de personal con grandes conocimientos informáticos para poder manejarlos.

#### 3.1.2 Analíticas web

Uno de los sistemas más comunes y empleados en la actualidad son las analíticas web. Estos sistemas proporcionan análisis de datos de una manera rápida y sencilla. Por el contrario su principal desventaja es que los datos a analizar vienen marcados con unas pautas predeterminadas, no existe mucha flexibilidad y solo podemos hacer lo que el sistema nos marca o proporciona, reduciendo así notablemente la capacidad de extracción de posible información útil.

#### 3.1.3 Sistemas expertos propios

Desarrollar un sistema experto propio por parte de una empresa no es lo habitual ya que estas prefieren suministrarse de sistemas ya hechos, pero aun así es una de las opciones a tener en cuenta. La desventaja principal de crear un sistema experto propio se basa en el coste y tiempo que se le ha de dedicar tanto a su desarrollo como a su mantenimiento, además solo el personal de esa empresa podrá tener nociones sobre su funcionamiento, cosa que implicara nuevamente un coste y tiempo en aprendizaje para cualquier persona externa que se incorpore a ella.

A su favor podemos decir que es un sistema totalmente privado y desmarcado de la competencia.

Los sistemas expertos pueden estar basados de tres maneras distintas, para conseguir la simulación del proceso de aprendizaje:

- En reglas , deterministas.

- En casos.
- En redes bayesianas, probabilísticos.

La estructura de la que estará compuesta un sistema experto es una base de conocimiento, una base de hechos (memoria), un motor de inferencia que es el que interpreta las reglas y datos y nos aporta una solución a ellos, módulos de justificación y una interfaz.

Los sistemas expertos están basados en muchos y variados algoritmos, como pueden ser el algoritmo básico de Markov, el algoritmo de Rete que es en el que están basados la mayoría como pueden ser Clips, JBoss Rules o BizTalk. También tenemos algunos como TAN, SAN, Naive Bayes que se basan en redes bayesianas y otros como Eager Evaluation, Lazy Evaluation, 1nn o C4.5.

#### **3.1.4 Microsoft Azure ML**

Microsoft Azure ML[1] es uno de los sistemas actuales capaces de estructurar, interpretar y hacer un aprendizaje automático de datos. Su funcionamiento se basa en ofrecernos una interfaz donde los usuarios pueden indicar concretamente qué quieren analizar y con qué condiciones.

Para que los datos sean analizados han de estar previamente ya estructurados y disponibles en el sistema de cloud de Microsoft, Azure. Para las empresas que tienen almacenados sus datos en Azure Compute Cloud de Microsoft, utilizar el sistema Microsoft Azure ML les supone una gran ventaja, debido a su fácil integración. Por el contrario Microsoft Azure ML puede no ser fácil de integrar en sistemas que no sean Microsoft, lo cual hace que no proporcione excesiva flexibilidad hacia otros sistemas.

#### **3.1.5 Google Prediction Api**

Google Prediction Api[2] es un sistema de interpretación y aprendizaje de datos con un enfoque muy similar a Microsoft Azure ML. Para poder utilizar este sistema necesitaremos tener almacenados los datos de nuestro proyecto en Google Cloud Platform. Este sistema utilizara una interfaz sencilla de cara al usuario donde pueda especificar que quiere predecir de los datos y hará las peticiones/request mediante OAuth2.0, recibiendo en un periodo muy corto de tiempo una respuesta a esa petición. Al contrario que Microsoft Azure ML, Google Prediction API tiene mayor flexibilidad a la hora de integrarse a cualquier sistema, ya que no tiene dependencia directa sobre ninguno.

#### **3.1.6 Big ML**

Big ML[3] es un sistema de predicción y aprendizaje de datos muy similar a los previamente explicados. Este sistema proporcionara al usuario una interfaz sencilla donde deberá introducir los datos, lo más estructurados posibles. Estos pueden ser introducidos directamente a la interfaz en prácticamente cualquier tipo de formato, o bien indicarle a esta la ubicación de los datos en lugares remotos, clouds. Una vez

efectuado este paso se indicara en qué condiciones se quiere hacer el aprendizaje, para posteriormente poder hacer las peticiones de predicción y que éstas sean lo más fiables posibles. Este sistema tiene la ventaja de que no es necesario tener creado ningún proyecto, ni depender de tener los datos en un cloud, podemos analizar ficheros independientes de datos que no estén relacionados a nada.

### **3.1.7 PredicSis**

PredicSis[4] es también un sistema de aprendizaje y predicción con muchas similitudes a los descritos anteriormente. Este sistema permite ser personalizado para las necesidades específicas que necesite el usuario, además está en un entorno de seguridad SaaS. Una característica a su favor es su alta escalabilidad, ya que nos proporcionan mediante su API una gestión y preparación de los datos que analizaremos de manera automática. Esto propicia una mejor integración en las aplicaciones.

### **3.1.8 IBM Watson Analytics**

IBM Watson Analytics[5] es un sistema de análisis y predicciones, que se basa en utilizar clouds para gestionar los datos de manera automática. El usuario especificara en una interfaz que situación desea analizar sobre estos y este generara resultados que el usuario podrá ir matizando y aclarando cada vez más, con el fin de concretar al máximo el análisis.

## **3.2 Conclusiones sistemas actuales**

Después de analizar los sistemas actuales de análisis e interpretación de datos se ha llegado a la conclusión de que en la fase de nuestro proyecto que trata el análisis e interpretación de los datos se utilizara un sistema ya existente, concretamente un sistema experto, que sea capaz de manera automatizada de interpretar y predecir los datos que previamente nosotros hayamos obtenido y estructurado.

De esta manera buscamos conseguir:

- Un sistema altamente escalable.
- Capacidad de encontrar personal cualificado para tratar con él, ya sea la propia empresa que desarrollara el proyecto o las empresas clientes que se favorecerán de él, cosa que hará ahorrar tiempos y costes.
- Mayor flexibilidad a futuros cambios, no hay dependencia a un solo sistema.
- Mayor flexibilidad al analizar los datos, se puede analizar cualquier cosa y no estamos limitados a pautas predeterminadas.
- Menor posibilidad de error al analizar los datos.

## 4 Formulación del problema

Disponer de información y datos fiables sobre cómo interactúan los usuarios en una web, y saber estructurar e interpretar esta información es una de las claves principales para tener éxito en el entorno empresarial.

Las empresas son conscientes de la importancia de disponer y analizar esta información para poder enfocar una estrategia correcta en como innovar en sus productos o hacer estos mejores y más eficientes, de cara a ofrecer a sus clientes o potenciales clientes aquello que solicitan y se requiere en el mercado actual. Manejar bien esta información puede suponer a una empresa ir un paso por delante de la competencia y por lo tanto tener mayor probabilidad de éxito. Es por ello que estas suelen invertir muchos recursos y esfuerzos en la obtención e interpretación de estos datos, pero esta tarea no es trivial ya que la cantidad de datos que se ha de manejar es inmensa y en la mayoría de casos este proceso es demasiado costoso y lento.

Por lo tanto, el objetivo principal de este proyecto es dar solución a la problemática anterior, y consiste en crear una REST API que es capaz de obtener, estructurar y distribuir toda la información e datos que nos interesen de una web. Esta tendrá que ser fácilmente reutilizable y escalable para cualquier portal web. Una vez se tengan estos datos de una manera estructurada, se distribuirán con el fin de que sean interpretados mediante un sistema experto, el cual será capaz de sacar conclusiones sobre los mismos, de tal manera que la empresa pueda tomar decisiones lo más fiables posibles sobre la estrategia a encarar de cara al futuro.

Todo queda documentado en este proyecto, desde la definición de objetivos, planificación temporal, el análisis de funcionalidades, hasta la presentación del prototipo final.

## 5 Alcance

El proyecto se centrará principalmente en conseguir un caso sencillo de obtención, estructuración y distribución de la información e datos de la interacción de un usuario con un portal web, e interpretar estos con un sistema experto que nos proporcionara conclusiones sobre ellos, y de esta manera conseguir cumplir los objetivos descritos anteriormente. Este deberá de estar correctamente documentado de tal manera que pueda ser fácilmente reutilizable y escalable en un futuro.

El alcance del proyecto será conseguir un sistema en el cual cada empresa (organización) pueda hacer un análisis de la información que deseen de la interacción de los usuarios por sus portales webs.

El proyecto se dividirá en tres escenarios:

- **Escenario 1: REST API**

La REST API será el elemento que nos proporcione las siguientes funcionalidades necesarias en el proyecto:

- Método POST:  
El método post es el que permitirá obtener la información de la interacción de los usuarios por un portal web, concretamente se obtendrá el identificador del usuario, la ruta por la cual ha navegado, y la fecha y hora de esa interacción.
- Método GET:  
El método get es el que permite estructurar la información obtenida de la interacción de los usuarios por un portal web, en un formato JSON/XML y mostrarla en una plataforma web.
- Distribución de datos:  
La información obtenida por el método post se distribuirá a una base de datos SQL de Amazon.

- **Escenario 2: Creación web**

Se creara un portal web de contenido, para hacer la navegación de los usuarios por un portal web.

- Página principal:  
En la página principal seleccionaremos la opción que queramos visitar.
- Paginas secundarias:

En las páginas secundarias es donde estará la información que hayamos seleccionado en las páginas previas.

- **Escenario 3: Interpretación y análisis de datos en un sistema experto**
  - Configuración datos:  
Se configurara y calibrara la información guardada en la base de datos, para obtener un mejor rendimiento de ellos en los siguientes pasos.
  - Configurar Sistema experto:  
Se configurara y calibrara el sistema experto con tal de conseguir una interpretación y análisis de los datos lo más fiables y exactas posibles, de cara a extraer las mejores conclusiones.

Mediante la implementación y configuración de las funcionalidades de los escenarios descritas previamente conseguiremos un sistema capaz de obtener, estructurar, distribuir e interpretar los datos e información de la interacción de un usuario con un portal web, de tal manera que las empresas puedan enfocar sus estrategias comerciales en función de las conclusiones extraídas con este sistema, cumpliendo de esta manera con los objetivos iniciales del proyecto.

## 5.1 Herramientas utilizadas

Las herramientas con las que se desarrollara este proyecto son:

### 5.1.1 Liferay 6.2 EE

Liferay 6.2 EE Portal[6] es un gestor de portales de código abierto que permite desarrollar portales webs complejos y personalizables de una manera rápida y eficiente. Este es uno de los más importantes en el mundo y fue creado en el año 2000. Este gestor, incluye un gestor de contenidos (CMS<sup>3</sup>), esta creado en Java y se utiliza para la creación de intranets privadas y portales webs. Una característica que facilita la creación de portales webs es que contiene más de 60 portlets (out of the box<sup>4</sup>) integrados.

Liferay ofrece un alto soporte de usuarios, escalabilidad, contenido multidioma y es un sistema multiplataforma responsive design<sup>5</sup>.

El entorno de desarrollo que está basado en Eclipse<sup>6</sup> y que se utilizara es Liferay Developer Studio, el cual utiliza Tomcat 7.0<sup>7</sup> como servidor local. El lenguaje de

---

<sup>3</sup> Es un programa que permite crear una estructura de soporte para la creación y administración de contenido web.

<sup>4</sup> Denominación a los componentes/funcionalidades predefinidos de Liferay.

<sup>5</sup> Diseño web visible y adaptable a varios dispositivos.

<sup>6</sup> Entorno de programación compatible con los lenguajes de programación Java, C, C++ y PHP.

<sup>7</sup> Servidor web basado en Java, con soporte para servlets y JSPs.



programación en el que se basa es Java<sup>8</sup>, además de utilizar también lenguajes como HTML y CSS.

#### **5.1.2 Bases de datos de Amazon, Amazon RDS**

La base de datos que utilizaremos para este proyecto, es una base de datos MySQL<sup>9</sup> de Amazon RDS<sup>10</sup>. Esta nos proporciona principalmente una gran capacidad de almacenamiento, totalmente escalable en función de las necesidades que se requieran.

Esta además se alinea perfectamente con los sistemas de Amazon necesarios para el aprendizaje de datos que utilizaremos y que se explican en los siguientes puntos.

#### **5.1.3 Sistema de almacenamiento, Amazon S3**

Amazon S3<sup>11</sup> nos proporcionara hacer almacenamientos parciales de nuestra base de datos, para hacer el posterior análisis de estos.

Esta además se alinea perfectamente con el sistema experto de Amazon que utilizaremos y que se explica en el siguiente punto.

#### **5.1.4 Sistemas expertos de Amazon, Amazon Machine Learning**

Utilizaremos sistemas expertos proporcionados por Amazon[7] para conseguir interpretar y analizar los datos obtenidos en los portales webs. En concreto utilizamos Amazon Machine Learning<sup>12</sup>, que nos permite crear modelos de aprendizaje y generar predicciones. Este sistema se compone de 3 pasos:

- El primer paso es el análisis de datos, en el cual se procesa y visualiza la distribución de estos con tal de sugerir y transformarlos, de tal manera que su formato sea el más optimizado de cara a extraer un mejor rendimiento de ellos en los siguientes pasos.
- El segundo paso, una vez transformados los datos en el paso anterior, se encarga de encontrar y guardar los patrones que haya detectado en ellos y que serán los que marcan el modelo predictivo.
- El último paso una vez encontrado los patrones de predicción, será evaluar y calibrar la precisión del modelo que se ha creado, de tal manera que la información obtenida sea lo más precisa y fiable posible.

Este sistema además proporciona una alta escalabilidad, tiene una alta integración con los diferentes sistemas proporcionados por Amazon lo cual hace que tenga una alta flexibilidad. También proporciona herramientas visuales como gráficos que ayudan a

---

<sup>8</sup> Lenguaje de programación.

<sup>9</sup> Sistema de gestión de base de datos relacional.

<sup>10</sup> Base de datos de Amazon.

<sup>11</sup> Sistema de almacenamiento de Amazon.

<sup>12</sup> Sistema experto de Amazon que realiza modelos de aprendizaje y predicciones sobre datos.

comprender la distribución de los datos y el contenido de ellos, así como a detectar posibles errores en los mismos.

## **5.2 Justificación de las tecnologías**

La razón principal por la cual se ha utilizado estas tecnologías, a parte de las mencionadas previamente, es por exigencia del cliente, que en este caso es la empresa dónde realizo las prácticas. Se ha de utilizar estas tecnologías para poder alinear al máximo este proyecto al desarrollo de la empresa, ya que todos sus proyectos se desarrollan con la tecnología Liferay y además utilizan en muchos de ellos Amazon Web Service como soporte para diferentes funcionalidades ya que los sistemas que proporciona este se alinean correctamente con las tecnologías de desarrollo de la empresa, ya que les facilita principalmente poder disponer de una alta escalabilidad, flexibilidad, seguridad y servicio técnico.

## **5.3 Obstáculos y planes de contingencia**

El obstáculo principal y más importante es el aprendizaje de las tecnologías utilizadas para desarrollar este proyecto. Liferay 6.2 EE es una tecnología bastante novedosa y a la cual se dedicara una gran parte de tiempo en el aprendizaje de la misma mediante documentación, tutoriales y pruebas.

Una de las ventajas en el aprendizaje de Liferay , que compensara el obstáculo que supone y que nos facilitara en cierta parte el trabajo, es el hecho de que utiliza principalmente el lenguaje de programación JAVA así como otros como HTML o CSS, que han sido lenguajes utilizados y practicados durante la carrera.

Otra tecnología utilizada en el desarrollo del proyecto son los sistemas de Amazon, Amazon RDS para la base de datos, Amazon S3, y Amazon Machine Learning para el sistema experto, los cuales también necesitaran una gran dedicación de tiempo en el aprendizaje mediante documentación, tutoriales y pruebas.

Para poder finalizar el proyecto en su periodo estimado es muy importante saber definir y planificar el tiempo que se le dedicara a las diferentes partes del proyecto, desde la parte de aprendizaje de las herramientas que se utilizaran hasta la parte de desarrollo, sabiendo identificar y priorizar que partes del mismo son más importantes.

## **6 Metodología y rigor**

El proyecto que realizare, es parte de un proyecto más grande que está realizando mi empresa y que tiene como destino un cliente real, por lo tanto este estará lo más alineado posible con las necesidades de esta. La metodología empleada en el proyecto estará basada en metodologías ágiles.

### **6.1 Metodología ágil**

Con la metodología ágil se proporcionaran entregas periódicamente. Estas entregas estarán estimadas en un periodo fijo de 1 semana, de esta manera tendremos un control periódico de resultados. Habrá reuniones semanales y comunicación constante con el responsable del proyecto y con mi grupo de trabajo con tal de compartir conocimientos y obtener feedback, para conseguir desarrollar las entregas en los términos y periodos establecidos, así como marcar los objetivos de la siguiente entrega una vez se haya entregado la que se estaba desarrollando y hacer una previsión de los riesgos que pueden surgir y afectar en el calendario pautado.

El hecho de tener feedback y seguimiento constante de mi proyecto, hará que pueda proporcionar mayor flexibilidad ante nuevas funcionalidades que puedan surgir.

### **6.2 Herramientas de seguimiento**

Las herramientas de seguimiento que se utilizaran es definir 3 listas, To Do, Doing y Done, que nos permitirán ubicar las tareas en cada una de ellas en función del estado de estas. Al mismo tiempo se realizaran reuniones diarias donde se informara el estado de las tareas que se están realizando al responsable del proyecto, así como acordar las tareas siguientes. Al final se hará un Review donde se mostraran todas las tareas realizadas y se determinara si cumplen con lo marcado con tal de poder seguir con las siguientes.

### **6.3 Método de validación**

Mediante la metodología ágil, se llevaran a cabo validaciones periódicas que nos permitirán tener un seguimiento de la evolución del proyecto. En cada reunión con la responsable y los miembros del equipo de trabajo se presentara las tareas realizadas, así como comentar los problemas e imprevistos que han surgido en la realización de las mismas y de esta manera poder tener una visión global del proyecto y poder definir mejor las tareas siguientes.

## **7 Planificación temporal inicial**

Después de definir el alcance del proyecto, se pasó a determinar la planificación inicial, organizando y estimando las tareas realizar.

### **7.1 Calendario**

EL tiempo estimado para realizar el proyecto es de 5 meses y 13 días. El proyecto se inició el día 11 de Enero de 2016 y se prevé que finalice el 24 de Junio de 2016. Se prevé destinar los últimos 12 días para completar la entrega final de la memoria, y la supervisión y corrección de la documentación, dando un margen de unos días desde el día 17 de Junio de 2016 hasta el día de la lectura marcada para la semana del 27 de Junio de 2016, con tal de realizar posibles correcciones y mejoras en la preparación de la lectura.

La dedicación de tiempo a la realización del proyecto por mi parte será de 20 horas semanales, en horario laboral de lunes a viernes de 9:00 a 13:00.

## 8 Iteraciones de desarrollo

El proceso de desarrollo de este proyecto se someterá a una metodología ágil. De esta manera el desarrollo se dividirá en 5 sprint<sup>13</sup> principales que tendrán una duración determinada y de los cuales se obtendrá una entrega de cada uno para su verificación. Estos sprint contendrán un conjunto de tareas más pequeñas que permitirán realizar una estimación de costes y tiempo más detallada y realista. En el apartado 10, Diagrama de Gantt, se puede ver con más profundidad y detalle los sprint y sus tareas.

A continuación se describen los 5 sprint a realizar:

### 8.1 Sprint Inception: Investigación, planificación y puesta en marcha

El Sprint Inception tiene como objetivo investigar cómo se realizara el proyecto y que herramientas serán las más apropiadas para llevarlo a cabo. También permitirá preparar el entorno de desarrollo así como hacer el aprendizaje y familiarizarse con las herramientas que se utilizaran.

Posteriormente se realizara la gestión del proyecto (GEP) en donde se estudiara y realizara el alcance y contextualización del proyecto, la planificación temporal, la gestión económica y la sostenibilidad del mismo.

El documento final de este sprint contendrá todo lo investigado y definido sobre lo que se realizara en el proyecto, así como el documento de gestión de proyectos (GEP), el cual será validado por el responsable durante el Review.

Esta iteración no tiene ninguna dependencia de precedencia.

### 8.2 Sprint 1: Implementación REST API y documentación

En el Sprint 1 llevaremos a cabo la definición de la estructura de la REST API, y la implementación de los diferentes métodos de la misma (GET, POST), así como hacer diferentes pruebas de su correcto funcionamiento. También se ira haciendo la memoria final del trabajo. Finalmente se realizara la validación del mismo.

Este sprint tiene una dependencia de precedencia que es el Sprint Inception definido en el punto 8.1 de este documento.

### 8.3 Sprint 2: Configuración sistema experto y documentación

En el Sprint 2 llevaremos a cabo la definición de las condiciones y normas que tendrá en cuenta nuestro sistema experto para poder hacer la interpretación y predicción de los datos. Una vez definido, llevaremos a cabo la configuración del sistema, indicándole las condiciones, sobre que datos y que tipo de estructuras. A continuación se harán diferentes pruebas para comprobar su correcto funcionamiento. También se

---

<sup>13</sup> Período de tiempo, en el cual se llevarán a cabo un conjunto de tareas.

ira haciendo la memoria final del trabajo. Finalmente se realizara la validación del mismo.

Este sprint tiene una dependencia de precedencia que es el Sprint 1, Implementación REST API, definido en el punto 8.2 de este documento.

#### **8.4 Sprint 3: Implementación web - REST API - sistema experto y documentación**

En el Sprint 3 llevaremos a cabo la definición de la estructura, que datos queremos obtener, distribuir e interpretar. Una vez definido, implementaremos la estructura del portal web y haremos la comunicación de los tres elementos, la comunicación web-REST API y REST API-sistema experto. A continuación se harán diferentes pruebas para comprobar el correcto funcionamiento. También se ira haciendo la memoria final del trabajo. Finalmente se realizara la validación del mismo.

Este sprint tiene una dependencia de precedencia que es el Sprint 2, Configuración sistema experto, definido en el punto 8.3 de este documento.

#### **8.5 Sprint 4: Documentación final**

En el Sprint 4 se llevara a cabo la elaboración de la finalización de la memoria del proyecto y la preparación de la lectura del TFG.

Este sprint tiene una dependencia de precedencia que es el Sprint 3, Implementación web -REST API – sistema experto, definido en el punto 8.4 de este documento.

## 9 Tabla de costes temporales

Tarea	Duración	Costes
<b>Sprint Inception</b>	11/01/2016 – 24/03/2016	216 horas
<b>Sprint 1</b>	25/03/2016 – 19/04/2016	72 horas
<b>Sprint 2</b>	20/04/2016 – 13/05/2016	72 horas
<b>Sprint 3</b>	16/05/2016 – 08/06/2016	72 horas
<b>Sprint 4</b>	09/06/2016 – 24/06/2016	48 horas
		<b>Total: 480 horas</b>

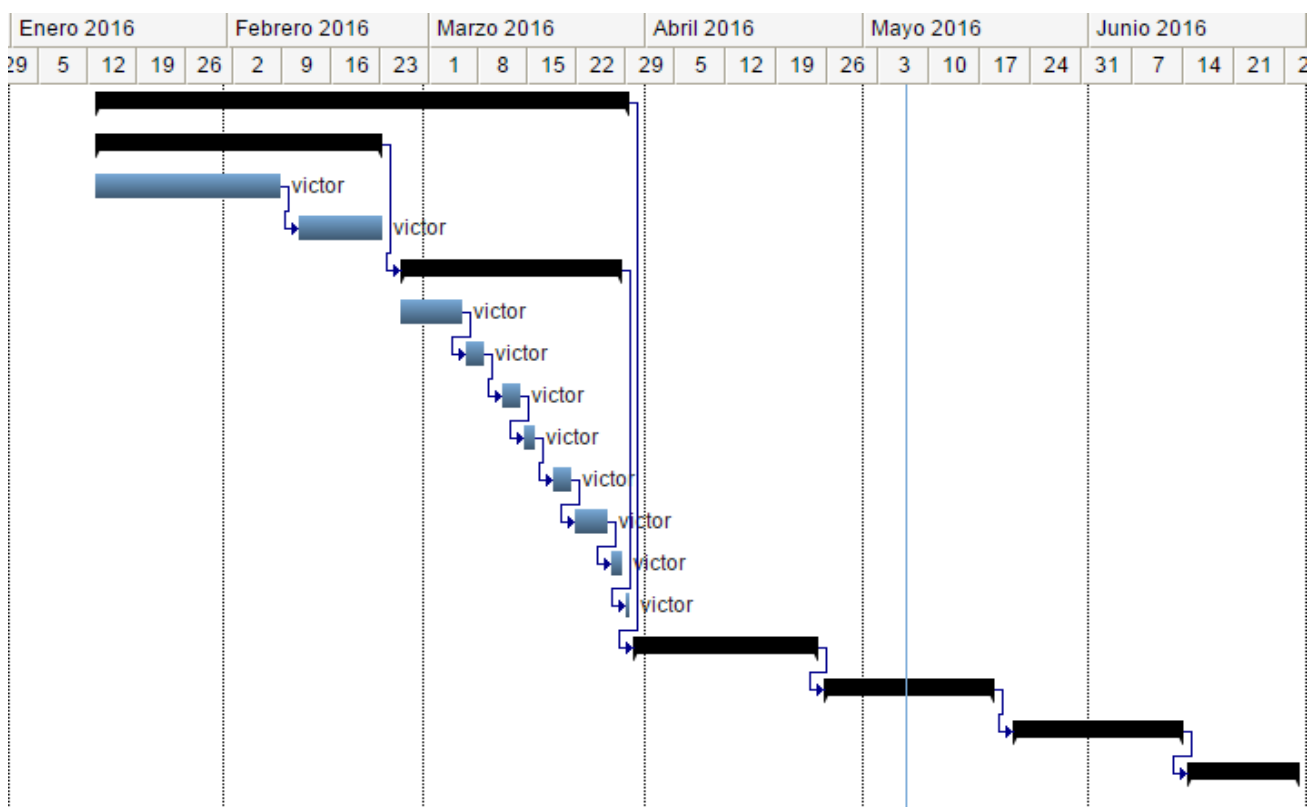
*Tabla 1: Tabla de costes de dedicación inicial*

## 10 Diagrama de Gantt

En los siguientes puntos mostraremos el diagrama de Gantt del proyecto. Lo subdividiremos en 2 partes para una mejor visión del mismo.

### 10.1 Parte Diagrama: Sprint Inception

	i	Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		☐ Sprint Inception: Investigación, planificación y puesta en marcha	54d?	11/01/2016	24/03/2016		
2		☐ Investigación previa	30d?	11/01/2016	19/02/2016		
3		Definición y preparación entorno de desarrollo	20d?	11/01/2016	05/02/2016		victor
4		Aprendizaje entorno de desarrollo	10d?	08/02/2016	19/02/2016	3	victor
5		☐ Gestión del proyecto (GEP)	23d?	22/02/2016	23/03/2016	2	
6		Alcance del proyecto y contextualización	7d?	22/02/2016	01/03/2016		victor
7		Planificación temporal	3d?	02/03/2016	04/03/2016	6	victor
8		Gestión económica y sostenibilidad	3d?	07/03/2016	09/03/2016	7	victor
9		Presentación preliminar	2d?	10/03/2016	11/03/2016	8	victor
10		Condiciones especialidad	3d?	14/03/2016	16/03/2016	9	victor
11		Preparación documento final	3d?	17/03/2016	21/03/2016	10	victor
12		Presentación	2d?	22/03/2016	23/03/2016	11	victor
13		Sprint Review	1d?	24/03/2016	24/03/2016	5	victor
14		☐ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
21		☐ Sprint 2: Configuración sistema experto y documentación	18d?	20/04/2016	13/05/2016	14	
27		☐ Sprint 3: Implementación web - REST API - sistema experto y documentación	18d?	16/05/2016	08/06/2016	21	
34		☐ Sprint 4: Documentación final	12d?	09/06/2016	24/06/2016	27	



**Figura 1: Diagrama de Gantt Sprint Inception inicial**



## 10.2 Parte Diagrama: Sprint 1, 2, 3, 4

	?	Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		☐ Sprint Inception: Investigación, planificación y puesta en marcha	54d?	11/01/2016	24/03/2016		
14		☐ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
15		Definición estructura REST API	3d?	25/03/2016	29/03/2016		victor
16		Implementación método POST	5d?	30/03/2016	05/04/2016	15	victor
17		Implementación método GET	5d?	06/04/2016	12/04/2016	16	victor
18		Testing	2d?	13/04/2016	14/04/2016	17	victor
19		Redacción memoria final	2d?	15/04/2016	18/04/2016	18	victor
20		Sprint Review	1d?	19/04/2016	19/04/2016	19	victor
21		☐ Sprint 2: Configuración sistema experto y documentación	18d?	20/04/2016	13/05/2016	14	
22		Definición sistema experto	3d?	20/04/2016	22/04/2016		victor
23		Configuración sistema experto	10d?	25/04/2016	06/05/2016	22	victor
24		Testing	2d?	09/05/2016	10/05/2016	23	victor
25		Redacción memoria final	2d?	11/05/2016	12/05/2016	24	victor
26		Sprint Review	1d?	13/05/2016	13/05/2016	25	victor
27		☐ Sprint 3: Implementación web - REST API - sistema experto y documentación	18d?	16/05/2016	08/06/2016	21	
28		Definición estructura	3d?	16/05/2016	18/05/2016		victor
29		Implementación web	5d?	19/05/2016	25/05/2016	28	victor
30		implementación web - REST API - sistema experto	5d?	26/05/2016	01/06/2016	29	victor
31		Testing	2d?	02/06/2016	03/06/2016	30	victor
32		Documentación memoria final	2d?	06/06/2016	07/06/2016	31	victor
33		Sprint Review	1d?	08/06/2016	08/06/2016	32	victor
34		☐ Sprint 4: Documentación final	12d?	09/06/2016	24/06/2016	27	
35		Redacción de la memoria	5d?	09/06/2016	15/06/2016		victor
36		Supervisión y corrección de la documentación	2d?	16/06/2016	17/06/2016	35	victor
37		Preparación de la lectura	5d?	20/06/2016	24/06/2016	36	victor

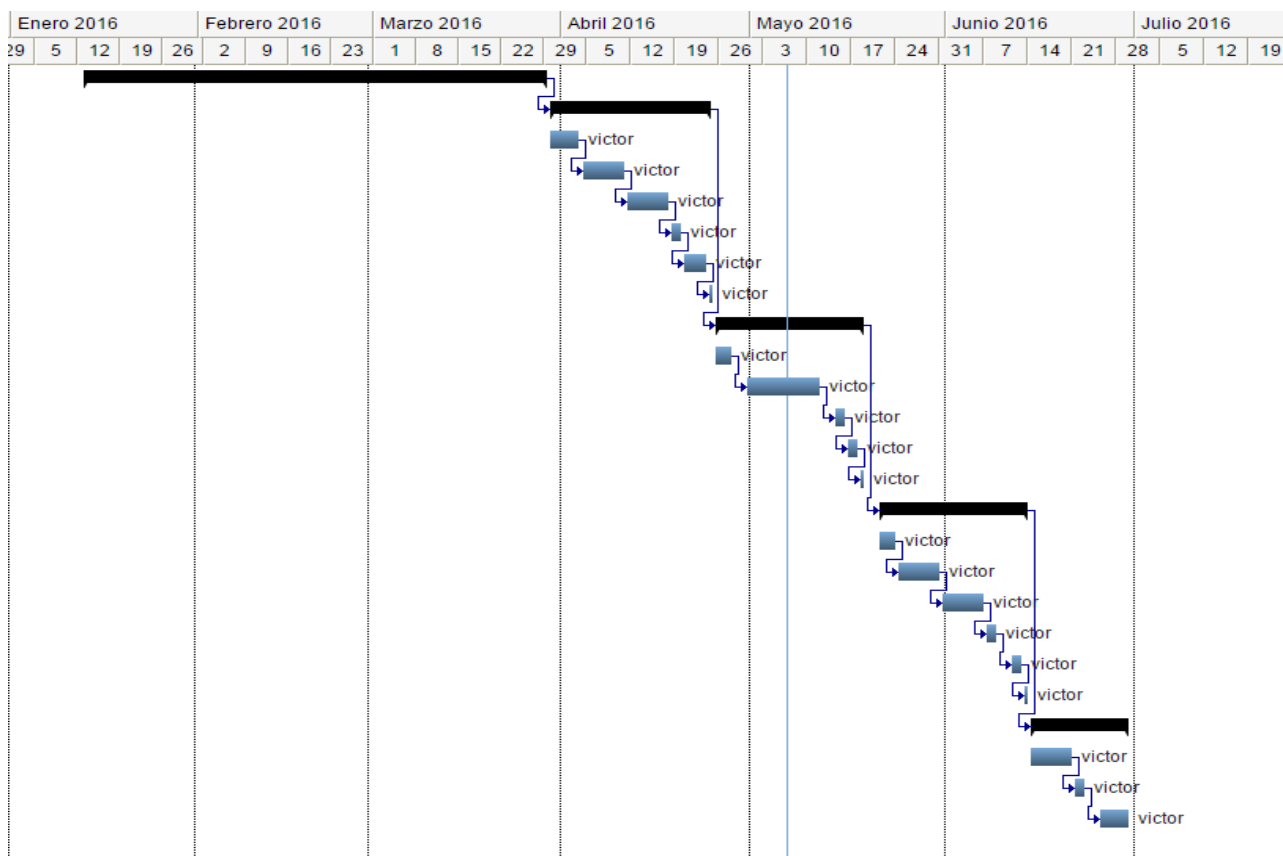


Figura 2: Diagrama de Gantt Sprint 1,2,3,4 inicial

## 11 Valoración de alternativas y plan de acción

A causa de la naturaleza de los proyectos de software en sí mismos, la planificación que se ha estimado puede tener desviaciones ya sean positivas o negativas en los diferentes Sprint. Si en alguno de los Sprint no se cumple con los objetivos marcados, se presentará el problema en el siguiente Sprint Review y se decidirá que es más prioritario, si el Sprint actual y por lo tanto se necesita reducir el tiempo de los siguientes Sprint o bien se priorizarán los siguientes Sprint.

Las tareas a reducir en cada Sprint en caso de que sea necesario para cumplir la planificación prevista serán: En el Sprint 1 se reducirá el número de métodos de la REST API, lo que supondrá reducir el tiempo de definición, implementación y testeo, lo que llevará a un ahorro de unas 20 horas (1 semana) en este Sprint. En el Sprint 2 se reducirá el número de condiciones y reglas a configurar en el sistema experto, lo que supondrá reducir el tiempo de definición, implementación y testeo, lo que llevará a un ahorro de unas 20 horas (1 semana) en este Sprint. En el Sprint 3 se reducirá el número de parámetros que se querrán obtener de la interacción de un usuario con la web, lo que supondrá reducir el tiempo de definición, implementación y testeo, lo que llevará a un ahorro de unas 20 horas (1 semana) en este Sprint. También disponemos de algún día de margen al final para poder alargar el Sprint 4 que es la documentación final y preparación de la lectura en caso de que sea necesario.

En cambio si las desviaciones de los Sprint son positivas ya que se han finalizado por debajo de su tiempo estimado, este tiempo se utilizaría para dedicárselo a futuros problemas que puedan surgir en los Sprint siguientes y por tanto conseguir de esta manera cumplir la estimación prevista. O por otra parte si no ha habido ningún tipo de problema, este tiempo se utilizaría para mejorar el producto que se está desarrollando.

## 12 Análisis económico inicial

Se ha realizado un estudio inicial sobre la viabilidad económica del proyecto, en el que se han analizado tanto los costes directos como los indirectos que están asociados al desarrollo del proyecto, así como un plan de contingencia en caso de desviación económica y un fondo de reserva para imprevistos. Por último se mostrara un resumen del presupuesto estimado total del coste del proyecto.

### 12.1 Costes directos por actividad

Los costes directos por actividad incluyen los recursos humanos involucrados en cada una de las etapas de inicio a fin del desarrollo del proyecto definidas en el diagrama de Gantt. Hemos de advertir que durante el TFG todas las actividades a los roles que se describirán a continuación seré yo quien las realice, pero para realizar el estudio de viabilidad económica se ha querido tener una visión de empresa donde cada actividad correspondería a un rol específico con sus respectivos costes actuales de mercado.

La remuneración<sup>14</sup> de los distintos roles, es la siguiente:

Rol	Remuneración anual (€)	Remuneración (€/h)
Jefe de proyecto	35000 €	18,23 €/h
Analista funcional	33000 €	17,19 €/h
Diseñador	18000 €	9,38 €/h
Programador	18000 €	9,38 €/h
Responsable de pruebas	22000 €	11,46 €/h

*Tabla 2: Remuneraciones salariales según el rol*

En la tabla siguiente mostramos los costes estimados de cada actividad del proyecto tal como está definido en el diagrama de Gantt.

Actividad	Unidades	Rol	Precio unitario(€/h)	Coste total (€)
Definición	40 h	Analista funcional	17,19 €/h	687,6 €
Preparación y aprendizaje entorno de desarrollo	80 h	Programador	9,38 €/h	750,4 €
Gestión del proyecto (GEP)	92 h	Jefe de proyecto	18,23 €/h	1677,16 €
Sprint Review	4 h	Jefe de proyecto	18,23 €/h	72,92 €
Sprint 1: Definición e implementación	52 h	Programador	9,38 €/h	487,76 €

<sup>14</sup> [http://www.pagepersonnel.es/sites/pagepersonnel.es/files/er\\_tecnologia16.pdf](http://www.pagepersonnel.es/sites/pagepersonnel.es/files/er_tecnologia16.pdf)

<b>Sprint 1: Testing</b>	8 h	Responsable de pruebas	11,46 €/h	91,68 €
<b>Sprint 1: Redacción memoria final</b>	8 h	Jefe de proyecto	18,23 €/h	145,84 €
<b>Sprint 1: Review</b>	4 h	Jefe de proyecto	18,23 €/h	72,92 €
<b>Sprint 2: Definición y configuración</b>	52 h	Programador	9,38 €/h	487,76 €
<b>Sprint 2: Testing</b>	8 h	Responsable de pruebas	11,46 €/h	91,68 €
<b>Sprint 2: Redacción memoria final</b>	8 h	Jefe de proyecto	18,23 €/h	145,84 €
<b>Sprint 2: Review</b>	4 h	Jefe de proyecto	18,23 €/h	72,92 €
<b>Sprint 3: Definición e implementación</b>	52 h	Programador	9,38 €/h	487,76 €
<b>Sprint 3: Testing</b>	8 h	Responsable de pruebas	11,46 €/h	91,68 €
<b>Sprint 3: Redacción memoria final</b>	8 h	Jefe de proyecto	18,23 €/h	145,84 €
<b>Sprint 3: Review</b>	4 h	Jefe de proyecto	18,23 €/h	72,92 €
<b>Sprint 4: Documentación final</b>	48 h	Jefe de proyecto	18,23 €/h	875,04 €
<b>Total</b>	<b>480 h</b>			<b>6457,72€</b>

*Tabla 3: Costes directos por actividad iniciales*

Además como coste directo se ha tenido también en cuenta tanto el hardware como el software empleado en el proyecto, ya que son indispensables para la realización del mismo.

#### **12.1.1 Amortización de la maquinaria**

La maquinaria empleada en este proyecto consta principalmente de un portátil DELL Latitude E5540, de un coste de 750€<sup>15</sup>. Este portátil, supondremos que tendrá una vida útil de 4 años y que el 80% del uso que le hagamos será destinado a este proyecto.

<sup>15</sup> <http://www.notebookcheck.net/Review-Dell-Latitude-E5540-Notebook.115887.0.html>

### 12.1.2 Amortización del software

Se utilizara la versión de pago del gestor de portales Liferay EE 6.2, el cual tiene un coste de 30000€ anuales. También se utilizaran infraestructuras virtuales, para poder hacer las predicciones de los datos el cual tendrá un coste de 100€ mensuales.

Producto	Unidades	Precio unitario	Dedicación	Coste estimado
Amortización de la maquinaria	5 meses	750 €/4 años	80%	62,5 €
Amortización del software	5 meses	31200 €/año	100%	13000 €
<b>Total</b>				<b>13062,5 €</b>

*Tabla 4: Costes directos maquinaria/software iniciales*

El coste directo total empleado será de 19250.22 €

	Coste estimado
<b>Roles</b>	6457,72€ €
<b>Maquinaria y software</b>	13062,5 €
<b>Total</b>	<b>19520,22 €</b>

*Tabla 5: Coste directo total inicial*

## 12.2 Costes indirectos

Se ha llevado a cabo un estudio de los recursos materiales y contrataciones de servicios necesarios para llevar hacia adelante este proyecto. A continuación serán explicados detalladamente:

### 12.2.1 Conexión de datos

Se utilizara una conexión a Internet de fibra óptica de 300MB con un coste de 36,18 €/mes<sup>16</sup>, de la cual un 50% de su ancho de banda se dedicara al proyecto.

### 12.2.2 Transporte

Se utilizara un abono mensual de 50 viajes, T-50-30, con un coste de 42,50 €.

### 12.2.3 Impresión en papel

Se consumirá papel principalmente en la entrega final de la memoria a cada uno de los miembros del tribunal y al director. Estimamos una extensión de aproximadamente 200 paginas, con un coste de 0,05 €/pagina.

<sup>16</sup> <http://www.movistar.es/empresas/para-tu-oficina/conectividad-internet/empresas-fibra-optica>

Producto	Unidades	Precio unitario	Dedicación	Coste estimado
Conexión de datos	5 meses	36,18 €/mes	50%	90,45 €
Transporte	5 meses	42,50 €/mes	100%	212,5 €
Impresión en papel	1200 paginas	0,05 €/pagina	100%	60 €
<b>Total</b>				<b>362,95 €</b>

*Tabla 6: Costes indirectos iniciales*

### 12.3 Contingencia

Se reserva una parte del presupuesto para la partida de contingencia, concretamente un 20% de la suma de los costes directos e indirectos.

Costes	Porcentaje	Coste de mercado	Coste de mercado
Directos	20%	19520,22 €	3904,05 €
Indirectos	20%	362,95 €	72,59 €
<b>Total</b>			<b>3976,64 €</b>

*Tabla 7: Costes de contingencia iniciales*

### 12.4 Imprevistos

#### 12.4.1 Retrasos

En el plan de acción de la planificación temporal se indica que se destina un margen de 20 horas (1 semana) durante el proyecto antes de la entrega final. Se estima que la probabilidad de entrar en un periodo de retardo es del 40% y lo contabilizaremos como horas de programador (9,38 €/h).

#### 12.4.2 Avería de la maquinaria

Se estima que la probabilidad de avería del equipo informático es de un 10%.

Imprevisto	Porcentaje	Unidades	Coste de mercado	Coste de mercado
Retrasos	40%	20 horas	9,38 €/h	75,04€
Avería de la maquinaria	10%	1 vez	750 €	75 €
<b>Total</b>				<b>150,04 €</b>

*Tabla 8: Costes de imprevistos iniciales*

## 12.5 Presupuesto

Concepto	Coste de mercado
Costes directos	19520,22 €
Costes Indirectos	362,95 €
Contingencia	3976,64 €
Imprevistos	150,04 €
<b>Total</b>	<b>24009,85 €</b>

*Tabla 9: Presupuesto final del proyecto inicial*

## 12.6 Control de gestión

Los costes calculados para los diferentes roles necesarios para este proyecto, son costes estimados ya que en este proyecto todos los roles serán ejercidos por mí mismo con una misma remuneración. No obstante se ha querido hacer una simulación real del proyecto en una empresa, para poder determinar la viabilidad del mismo.

Se mantendrá un registro de las horas realizadas por tarea para comprobar que realmente se sigue lo establecido en el diagrama de Gantt. En cada Sprint Review se tendrá en cuenta este registro de horas para poder identificar posibles desviaciones y poderlas rectificar para ajustarse a la planificación temporal. Al final de la entrega se realizara también un resumen real de horas destinadas y costes para compararlos con los de la planificación, y poder ver de esta manera los porcentajes de desviación que ha habido y corregirlos de cara a futuros proyectos. Para todo ello se tendrán en cuenta los siguientes indicadores de desviación, para llevar un control más detallado sobre la gestión del proyecto:

- Desviación horas tarea = horas estimadas subtareas - horas real subtareas.
- Desviación horas Sprint = horas estimadas tareas - horas real tareas.
- Desviación horas = horas estimadas Sprints - horas real Sprints.
- Desviación costes tarea = costes estimados subtareas - costes real subtareas.
- Desviación costes Sprint = costes estimados tareas - costes real tareas.
- Desviación costes = costes estimados Sprints - costes real Sprints.

## 13 Sostenibilidad

A continuación mostramos el estudio sobre sostenibilidad realizado en este proyecto, dando explicación de cada uno de sus puntos en los siguientes apartados.

En las tres siguientes tablas explicaremos cómo funciona el rango de puntuaciones dado para cada apartado.

	0 Puntos	10 Puntos
<b>Planificación, desarrollo e implementación (PPP)</b>	Nada sostenible	Muy sostenible

*Tabla 10: Puntuación planificación, desarrollo e implementación*

	0 Puntos	20 Puntos
<b>Vida Útil</b>	Nada sostenible	Muy sostenible

*Tabla 11: Puntuación vida útil*

	-20 puntos	0 puntos
<b>Riesgos</b>	Mucho riesgo	Nada de riesgo

*Tabla 12: Puntuación Riesgos*

En la siguiente tabla mostramos las puntuaciones dadas para cada apartado.

	PPP(0-10)	Vida Útil(0-20)	Riesgos(-20-0)
<b>Ambiental</b>	8	18	-2
<b>Económico</b>	7	15	-5
<b>Social</b>	9	18	-5
<b>Rango sostenibilidad parcial</b>	24	51	-12
<b>Total</b>			<b>63</b>

*Tabla 13: Puntuaciones parciales y total de sostenibilidad del proyecto*

La puntuación total que sale es la suma de los valores del rango de sostenibilidad parcial y es de 63 puntos sobre 90 puntos posibles, por lo tanto se considera un resultado óptimo para lo que ofrece y propone nuestro proyecto, ya que es económicamente viable para las empresas, su impacto social es positivo para los usuarios, empresas y mi persona, y a nivel medioambiental no genera valores negativos.



A continuación explicamos el razonamiento para dar esta puntuación a cada subapartado.

### **13.1 Ambiental**

Este proyecto no ha de causar ningún tipo de daño al medioambiente, por lo tanto no genera ningún valor negativo sobre este.

No obstante se genera algún tipo de consumo eléctrico en el espacio de trabajo como podría ser el del consumo del portátil, internet, hosting o también el CO2 que se genera del transporte público utilizado, pero será considerado mínimo y por tanto podemos decir que durante la etapa de planificación, desarrollo e implantación del proyecto (PPP) este proyecto es bastante sostenible, poniéndole una puntuación de 8 sobre 10.

A lo largo de su vida útil, este proyecto prácticamente en su totalidad será reutilizable para futuros proyectos dentro de la empresa que se dediquen al análisis de datos, lo cual implicara no tener ningún tipo de gasto extra en recursos materiales y eléctricos. Con lo cual a nivel de consumo energético tendrá un gasto similar al de su fase de PPP. También no se prevé el desmantelamiento de ningún recurso material al acabar el proyecto ya que seguirán siendo utilizados, ni se necesitaran materias primas ni nuevos productos. Por todo ello la puntuación del aspecto ambiental durante su vida útil será de 18.

A nivel de riesgos, este proyecto tiene pocos, el mayor de ellos sería una desviación en los tiempos de los mismos y que se alargara el proceso de desarrollo e implantación cosa que implicaría en un mayor gasto eléctrico. Pero en este caso la variación podría ser mínima, por lo tanto la puntuación será de -2.

### **13.2 Económica**

El coste total del proyecto calculado y en el que se han tenido en cuenta costes directos, indirectos, contingencia e imprevistos es de unos 24009,85 €. Este es un precio que muchas empresas podrían permitirse, pero no todas, por lo tanto su puntuación es de 7 sobre 10.

Este coste permite que el proyecto sea viable y competitivo ya que es un sistema reutilizable y que puede ser utilizado a largo plazo, lo cual hace que la inversión sea fácilmente recuperada, por lo tanto la nota es de 15.

Este proyecto además, a nivel de riesgos, aunque existe la posibilidad de que se realicen proyectos similares, difícilmente se podrá realizar con un coste más bajo debido a que en el tiempo y recursos empleados en él ya se ha tenido en cuenta la optimización al máximo de estos. En este precio se han tenido en cuenta todos los posibles costes, ya sean humanos, materiales, de imprevistos y contingencias. Por todo ello este proyecto a 5 meses se puede considerar viable si se cumple la planificación

temporal y de costes prevista. Un factor de riesgo a tener muy en cuenta es el coste de disponer de la licencia de Liferay EE, ya que tiene un coste de 30000€ y es un pago anual, lo cual aumenta considerablemente el presupuesto final, lo cual implica una pérdida potencial de empresas que se pueden permitir este proyecto.

Como alternativa, se puede considerar la utilización de la versión gratuita de Liferay, Liferay CE, lo que nos permitiría reducir los costes considerablemente y hacer que más empresas se lo puedan permitir, pero también se reduce la calidad del producto. Por todo ello la puntuación final es de -5.

### **13.3 Social**

Este proyecto ha tenido una afectación muy positiva en mi persona, ya que me ha permitido adquirir conocimientos nuevos, así como aplicar y perfeccionar conocimientos que había ido adquiriendo a lo largo de la carrera, ya sean a nivel práctico o teóricos. Todos estos conocimientos me servirán de gran ayuda en mi presente y futuro laboral, por lo tanto la puntuación dada es de 9.

El objetivo principal de este proyecto es ofrecer un sistema en el que se pueda obtener, estructurar, distribuir y predecir e analizar los datos que se obtienen de la interacción de un usuario por una web. En la actualidad saber analizar grandes cantidades de información se ha convertido en una de las prioridades a nivel empresarial, pero pocas empresas disponen de un sistema eficiente para ello. Es por ello que este proyecto, tiene la intención de cubrir estas necesidades, creando un sistema capaz de interpretar estos datos. A nivel empresarial les reportara tener una mayor conciencia sobre lo que quieren o buscan sus usuarios, clientes o potenciales clientes de ellos y poder tomar decisiones al respecto. De esta manera la empresa seguramente podrá ofrecer a los usuarios lo que estos demandan en cada momento, lo cual implicara no solo en la empresa sino en la satisfacción de los usuarios. Por tanto este proyecto podría beneficiar a cualquier tipo de empresa, y cuantos más usuarios o clientes, más beneficio se sacara del sistema. Por todo ello la puntuación de este subapartado es de 18.

Además este proyecto, a nivel de riesgos, no supone grandes amenazas para ningún tipo de colectivo. Un factor de riesgo a tener en cuenta podría ser la no aceptación por parte de los usuarios a que se obtengan datos de sus interacciones por una web, pero el hecho de que estos datos sean anónimos y la sociedad actual este ya muy adaptada a que sus datos sean utilizados como base de información, hace que el riesgo en este caso sea considerado mínimo. Por todo ello la puntuación en este subapartado es de -5.

## 14 Análisis de requisitos

En este apartado se definirán los requisitos funcionales del sistema y los no funcionales. En este sistema hay dos tipos de usuarios:

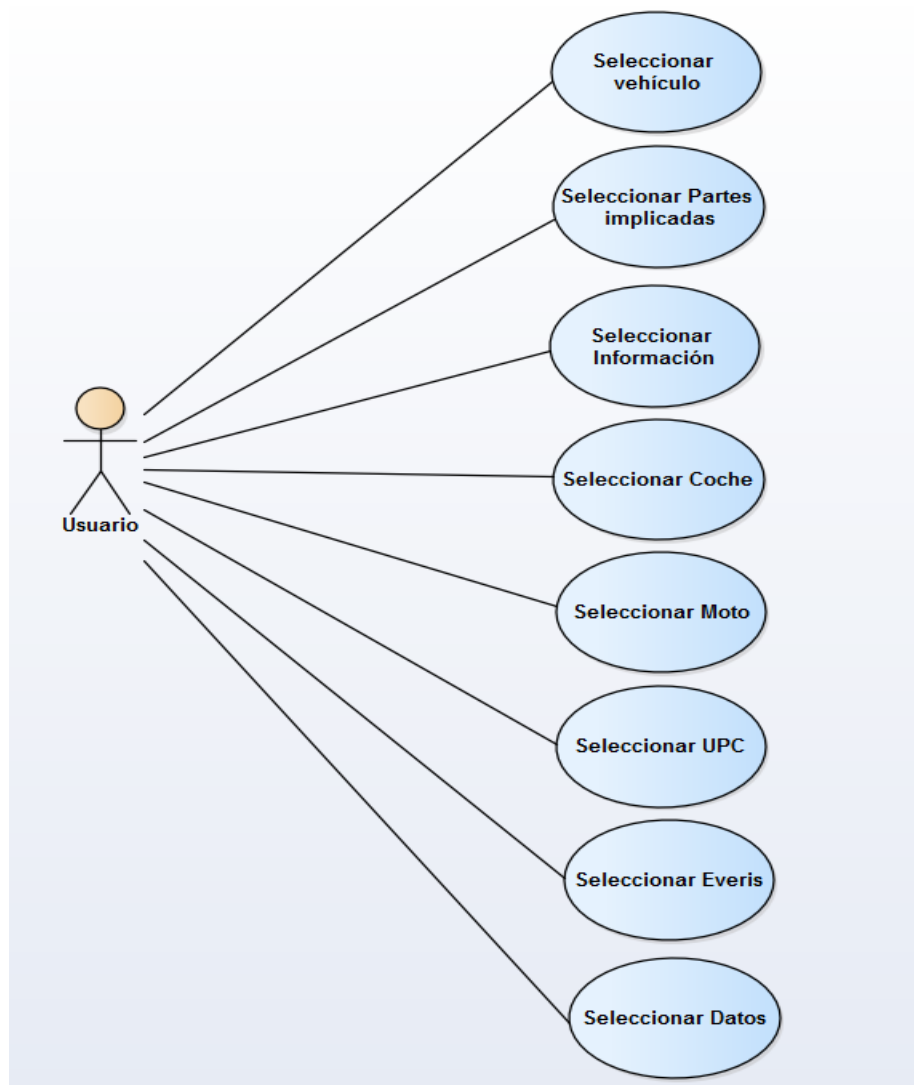
- Los usuarios que serán administradores del sistema, y que se encargaran de todo el control sobre la base de datos, así como el análisis de esta.
- Los usuarios que interactúan por el portal web.

### 14.1 Requisitos funcionales

Para definir las acciones o comportamiento del sistema utilizaremos los requisitos funcionales. Concretamente definiremos los requisitos funcionales utilizados en este sistema, separándolos en dos tipos de usuarios, administrador/sistema y usuario.

#### 14.1.1 Casos de Uso WEB

##### 14.1.1.1 Diagrama



*Figura 3 : Diagrama de casos de uso web*

#### **14.1.1.2 Seleccionar Vehículo**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de estar en la página principal.
<b>Descripción</b>	El usuario podrá ver qué tipo de vehículos hay disponibles, seleccionando la opción Vehículos.
<b>Casos de error</b>	-

#### **14.1.1.3 Seleccionar Partes implicadas**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de estar en la página principal.
<b>Descripción</b>	El usuario podrá ver qué organizaciones o empresas están involucradas en este proyecto, seleccionando la opción Partes implicadas.
<b>Casos de error</b>	-

#### **14.1.1.4 Seleccionar Información**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de estar en la página principal.
<b>Descripción</b>	El usuario podrá ver qué tipo de información hay disponible, seleccionando la opción Información.
<b>Casos de error</b>	-

#### **14.1.1.5 Seleccionar Coche**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de seleccionar previamente la opción Vehículo.
<b>Descripción</b>	El usuario podrá ver información sobre coches, seleccionando la opción Coche.
<b>Casos de error</b>	-

#### **14.1.1.6 Seleccionar Moto**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de seleccionar previamente la opción Vehículo.
<b>Descripción</b>	El usuario podrá ver información sobre motos, seleccionando la opción Moto.
<b>Casos de error</b>	-

#### **14.1.1.7 Seleccionar UPC**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de seleccionar previamente la opción Partes implicadas.
<b>Descripción</b>	El usuario podrá ver información sobre la Universidad Politécnica de Cataluña, seleccionando la opción UPC.
<b>Casos de error</b>	-

#### **14.1.1.8 Seleccionar Everis**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de seleccionar previamente la opción Partes implicadas.
<b>Descripción</b>	El usuario podrá ver información sobre Everis, seleccionando la opción Everis.
<b>Casos de error</b>	-

#### **14.1.1.9 Seleccionar Datos**

<b>Actores</b>	Usuario.
<b>Precondición</b>	El usuario ha de seleccionar previamente la opción Información.
<b>Descripción</b>	El usuario podrá ver información genérica, seleccionando la opción Datos.
<b>Casos de error</b>	-

### 14.1.2 Casos de Uso REST API

#### 14.1.2.1 Diagrama

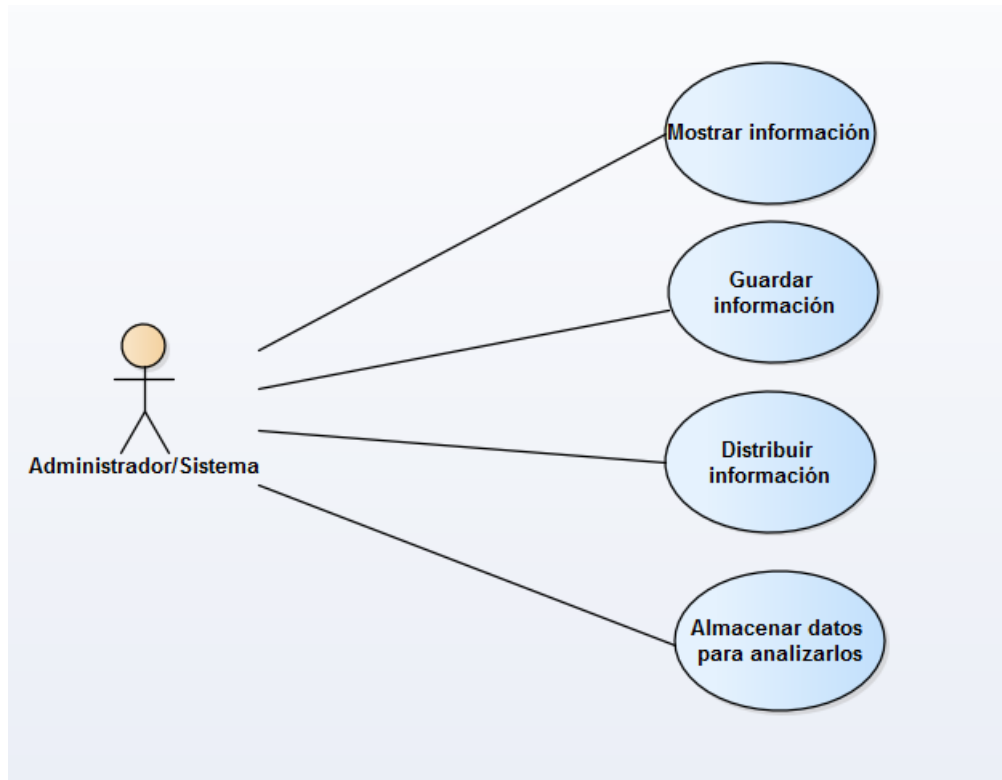


Figura 4 : Diagrama de casos de uso REST Api

#### 14.1.2.2 Mostrar información

<b>Actores</b>	Administrador.
<b>Precondición</b>	El administrador ha de saber la URL correspondiente al método GET.
<b>Descripción</b>	El administrador podrá consultar la información obtenida de la interacción de los usuarios por el portal web, concretamente el identificador y las rutas visitadas por cada usuario. Para ello deberá introducir la URL correspondiente del método GET. Esta información será mostrada en formato XML.
<b>Casos de error</b>	La dirección URL introducida no es correcta o no ha sido introducida.

#### 14.1.2.3 Guardar información

<b>Actores</b>	Sistema.
<b>Precondición</b>	El usuario ha de haber visitado una de las rutas marcadas como objetivo.

<b>Descripción</b>	El sistema guardara el identificador de sesión del usuario que esta interactuando en el portal web y la ruta marcada como objetivo que ha visitado.
<b>Casos de error</b>	-

#### **14.1.2.4 Distribuir información**

<b>Actores</b>	Sistema.
<b>Precondición</b>	El usuario ha de haber visitado una de las rutas marcadas como objetivo.
<b>Descripción</b>	El sistema distribuirá la información obtenida de la interacción del usuario por un portal web, concretamente el identificador, la ruta marcada como objetivo que ha visitado, la fecha y hora de la interacción y además asignara un estado inicial a 0 para marcar que esos datos no han sido aún asignados para el análisis con el sistema experto, mediante el método POST a la base de datos SQL de Amazon RDS.
<b>Casos de error</b>	-

#### **14.1.2.5 Almacenar datos para analizarlos**

<b>Actores</b>	Administrador.
<b>Precondición</b>	El administrador ha de saber la URL correspondiente al método almacenar.
<b>Descripción</b>	El administrador podrá almacenar en Amazon S3 aquellos datos de la base de datos SQL de Amazon RDS que aún no hayan sido seleccionados para ser analizados en el sistema experto, clicando sobre el botón Guardar.
<b>Casos de error</b>	La dirección URL introducida no es correcta o no ha sido introducida.

### 14.1.3 Casos de Uso Sistemas Amazon

#### 14.1.3.1 Diagrama

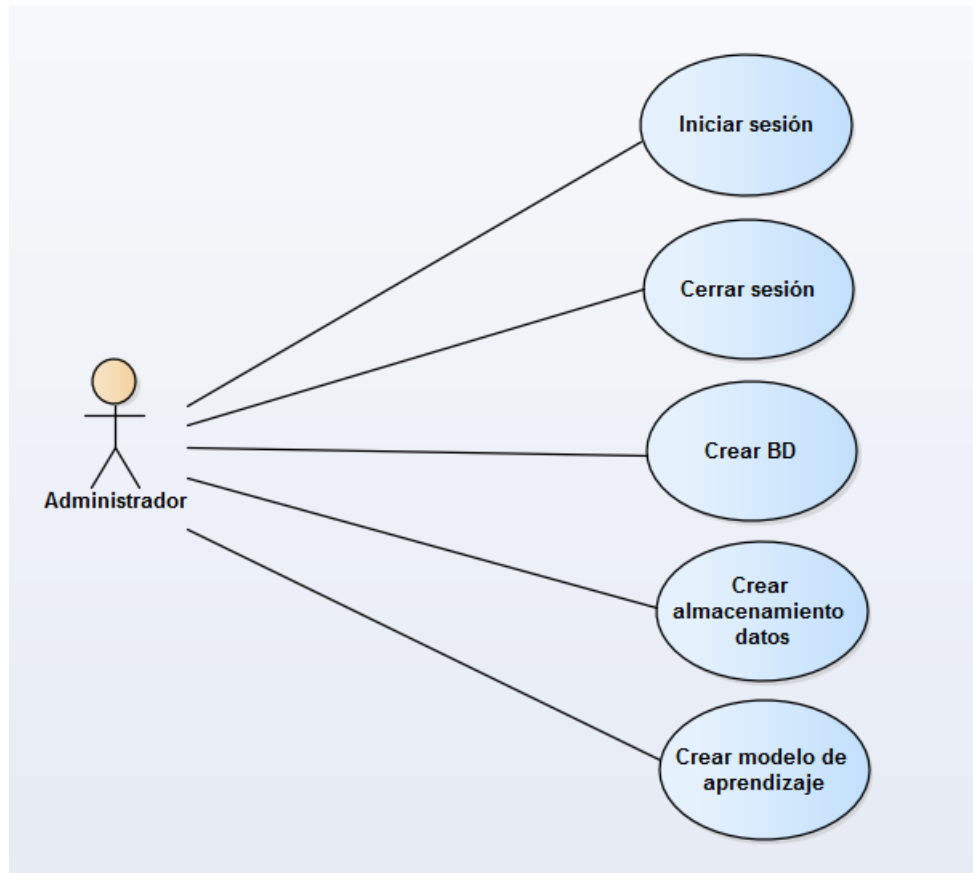


Figura 5 : Diagrama de casos de uso sistemas Amazon

#### 14.1.3.2 Iniciar sesión

<b>Actores</b>	Administrador.
<b>Precondición</b>	El usuario ha sido registrado anteriormente.
<b>Descripción</b>	<p>El usuario podrá iniciar sesión en Amazon, accediendo a <a href="https://aws.amazon.com/es/">https://aws.amazon.com/es/</a> y clicando en el botón Inscribirse, acto seguido el usuario deberá completar el formulario indicando e-mail y contraseña.</p> <p>Una vez iniciada sesión, el usuario será redireccionado a la página principal de Amazon, donde dispondrá de todos los sistemas disponibles.</p>
<b>Casos de error</b>	<p>La dirección URL introducida es incorrecta.</p> <p>El correo electrónico y/o contraseña introducidos son incorrectos.</p>



#### **14.1.3.3 Cerrar sesión**

<b>Actores</b>	Administrador.
<b>Precondición</b>	El usuario tiene la sesión iniciada.
<b>Descripción</b>	El usuario podrá cerrar sesión clicando el botón de Log Out/Cerrar sesión. Una vez cerrada la sesión, el usuario será redireccionado a la página inicial de Amazon, donde podrá volver a iniciar sesión si lo desea.
<b>Casos de error</b>	-

#### **14.1.3.4 Crear BD**

<b>Actores</b>	Administrador.
<b>Precondición</b>	El usuario tiene la sesión iniciada.
<b>Descripción</b>	El usuario podrá crear una base de datos SQL clicando sobre Amazon RDS. Una vez clicado el usuario será redireccionado al panel de control de Amazon RDS donde podrá crear una instancia/base de datos clicando sobre el botón DB Launch Instance, donde a continuación se mostrara un formulario que el usuario deberá completar para indicar que condiciones sobre esa base de datos desea.
<b>Casos de error</b>	-

#### **14.1.3.5 Crear almacenamiento datos**

<b>Actores</b>	Administrador.
<b>Precondición</b>	El usuario tiene la sesión iniciada.
<b>Descripción</b>	El usuario podrá crear un sistema de almacenamiento clicando sobre Amazon S3. Una vez clicado el usuario será redireccionado al panel de control de Amazon S3 donde podrá crear un bucket clicando sobre el botón Create Bucket, donde a continuación se mostrara un formulario que el usuario deberá completar para crearlo.
<b>Casos de error</b>	-

#### **14.1.3.6 Crear modelo de aprendizaje**

<b>Actores</b>	Administrador.
<b>Precondición</b>	El usuario tiene la sesión iniciada.
<b>Descripción</b>	El usuario podrá crear un modelo de aprendizaje para hacer un análisis de datos clicando sobre Amazon Machine Learning. Una vez clicado el usuario será redireccionado al panel de control de Amazon Machine Learning donde podrá crear un modelo de aprendizaje de donde se hará el análisis sobre estos.
<b>Casos de error</b>	-

### **14.2 Requisitos no funcionales**

Los requisitos no funcionales, son todos los que permiten el correcto funcionamiento del software y por lo tanto que este se considere un éxito. Estos requisitos tendrán en cuenta requisitos de calidad del software como la fiabilidad, la eficiencia y la seguridad.

A continuación se detallan los requisitos no funcionales que el sistema en todo momento cumple y la metodología utilizada para conseguirlo:

#### **14.2.1 Fácil de utilizar para cualquier tipo de usuario**

El sistema es fácil de utilizar de cara al usuario que interactúa por la web, ya que la interfaz del sistema está bien definida y es intuitiva, con lo cual no es necesario requerir de grandes conocimientos técnicos sobre informática.

Por lo que respecta al administrador ha de disponer de un conocimiento técnico básico sobre informática para trabajar con elementos como la base de datos o el sistema experto, aunque el manejo de estas también está correctamente definido y por lo tanto es fácil de utilizar.

#### **14.2.2 Reutilizable y personalizable para cualquier empresa e organización**

Este sistema está diseñado para que cualquier empresa e organización que disponga de un portal web pueda hacer uso de él. Además la empresa podrá determinar que datos e información desea obtener de la interacción de los usuarios por sus portales webs, para posteriormente analizarla según también los parámetros de configuración que esta desee.

#### **14.2.3 Garantiza la seguridad de la información**

Para poder tratar y analizar los datos, el administrador tendrá que estar registrado con una cuenta de Amazon Web Service que solo tendrá acceso a estos. Además todos los datos obtenidos serán anónimos para preservar la identidad de los usuarios que interactúan por los portales webs.

#### ***14.2.4 Buena apariencia***

El sistema ofrece una interfaz web amigable, debido a la aplicación de plantillas, la aplicación de CSS y la unificación de colores. También la REST API ofrece una visualización correcta mediante el formato XML.

#### ***14.2.5 Alta disponibilidad y escalabilidad***

El sistema ofrece una alta disponibilidad, ya que todos los datos obtenidos se alojaran y serán tratados en una base de datos SQL de Amazon Web Service. Además también nos proporciona una alta escalabilidad debido a su facilidad para incrementar la capacidad de la base de datos.

#### ***14.2.6 Rápido***

El sistema carga las páginas y guarda los datos de manera rápida y fluida, debido a su correcta implementación.

#### ***14.2.7 Usable***

El sistema ofrece a las empresas e organizaciones la capacidad de hacer un mejor análisis de la información obtenida por sus portales webs, permitiendo de esta manera que el sistema sea fácilmente usable.

#### ***14.2.8 Iconos convencionales***

El sistema utiliza iconos intuitivos, fáciles de interpretar para cualquier tipo de usuario.

#### ***14.2.9 Actualizaciones***

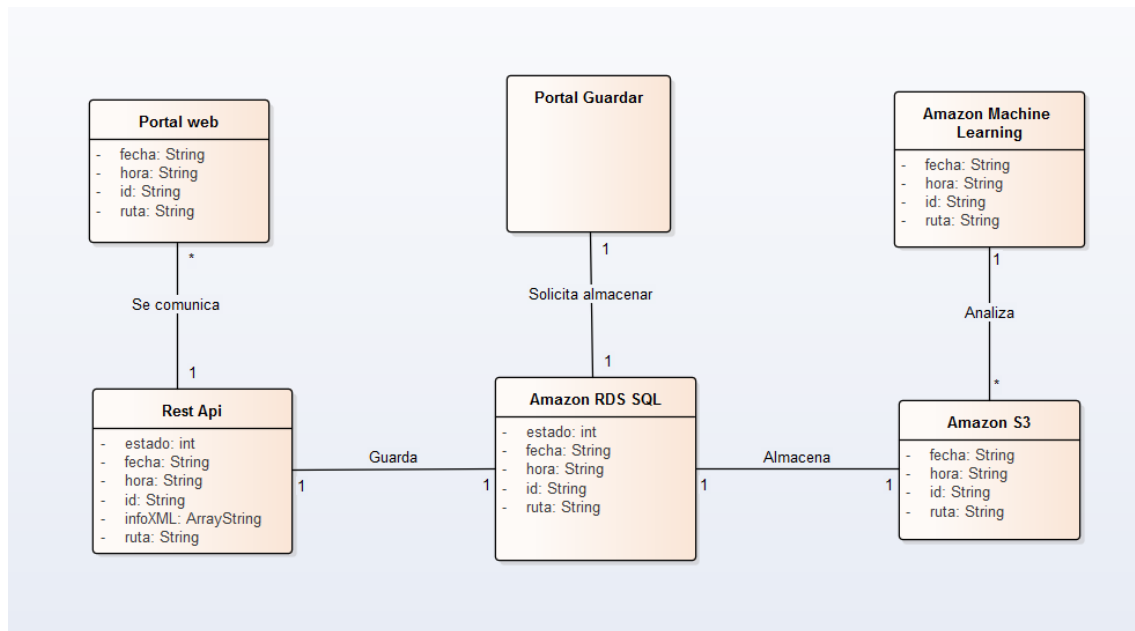
Tanto Liferay como Amazon ofrecen actualizaciones constantes, lo cual hace que el sistema pueda mantener de manera constante su buen funcionamiento.

#### ***14.2.10 Soporte constante***

Tanto Liferay como Amazon disponen de un servicio de soporte en caso de problemas con estos. Además Liferay dispone de un foro oficial público de cara a poder solucionar cualquier problema o imprevisto surgido.

## 14.3 Diagramas

### 14.3.1 Diagrama Modelo conceptual de datos



**Figura 6 : Diagrama modelo conceptual de datos**

### 14.3.2 Descripción Modelo conceptual de datos

El modelo conceptual de datos está basado en primer lugar de un portal web, el cual tendrá 4 datos principales que serán el identificador de la sesión, la ruta por la cual ha navegado, y la fecha y hora de esa navegación. Este se comunicara con una Rest Api la cual tendrá todos los parámetros del portal web así como un array que contendrá toda esta información, y un nuevo campo, estado, para controlar que cada interacción de un usuario por la web aún no ha sido almacenada con tal de que la trate el sistema experto.

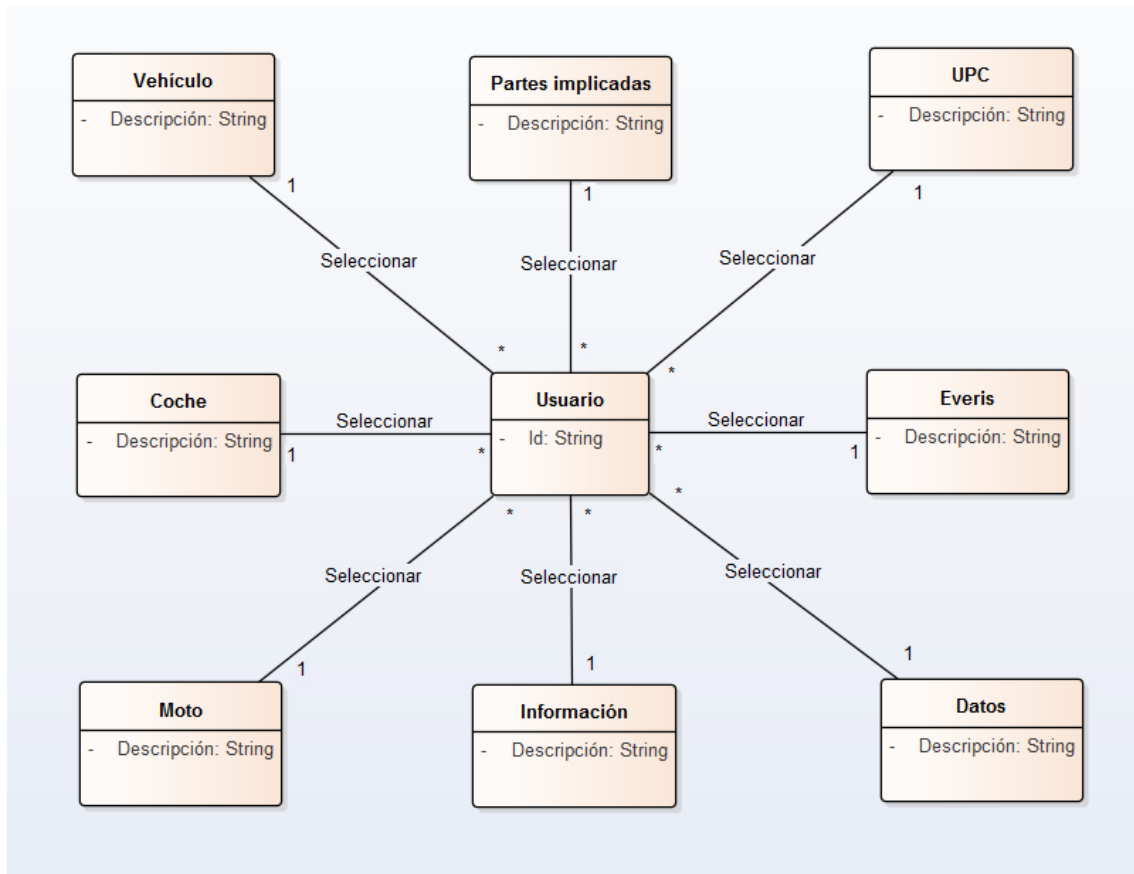
La Rest Api guarda en nuestra base de datos de Amazon RDS SQL, el identificador, ruta, fecha, hora y estado de esa interacción por el portal web.

El Portal Guardar, solicita el almacenamiento de los datos aun no almacenados en Amazon S3.

Amazon RDS SQL almacena en nuestro lugar de almacenamiento de Amazon S3 y cambia el valor del parámetro estado de los datos. Amazon S3 contendrá el identificador, ruta, fecha y hora.

Por ultimo Amazon Machine Learning analizara estos datos.

### 14.3.3 Diagrama UML del portal web



**Figura 7 : Diagrama UML portal web**

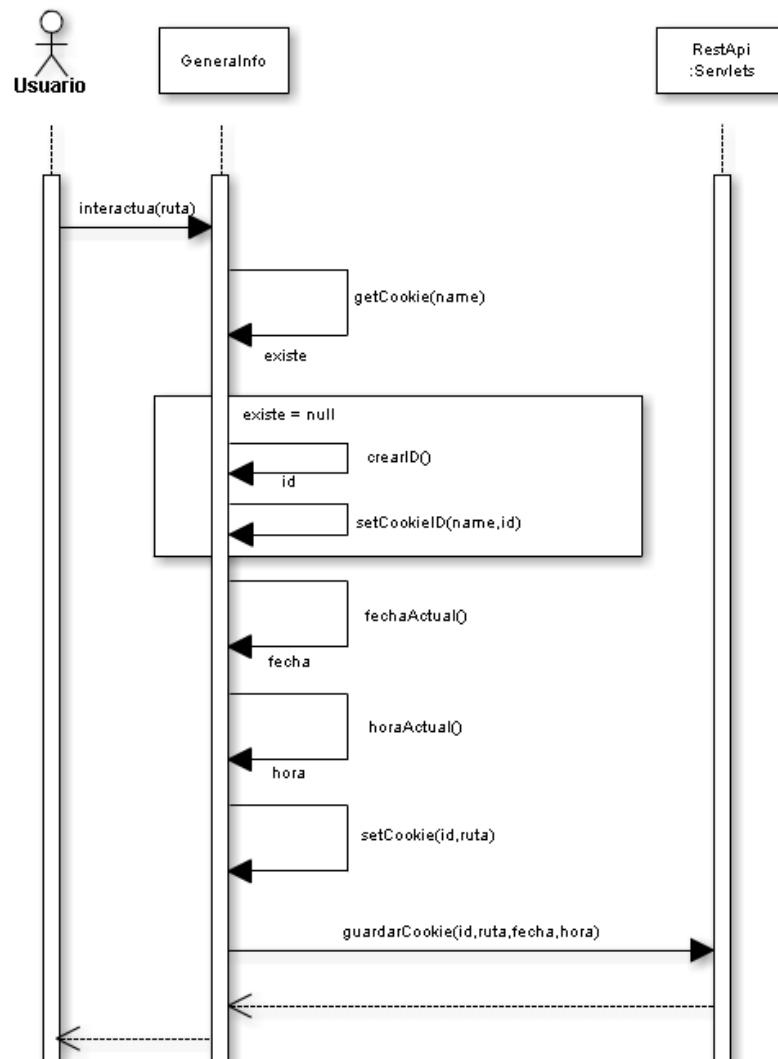
### 14.3.4 Descripción UML del portal web

En este diagrama UML<sup>17</sup> tenemos un Usuario, el cual contendrá un identificador. Este puede seleccionar cualquiera de las opciones disponibles.

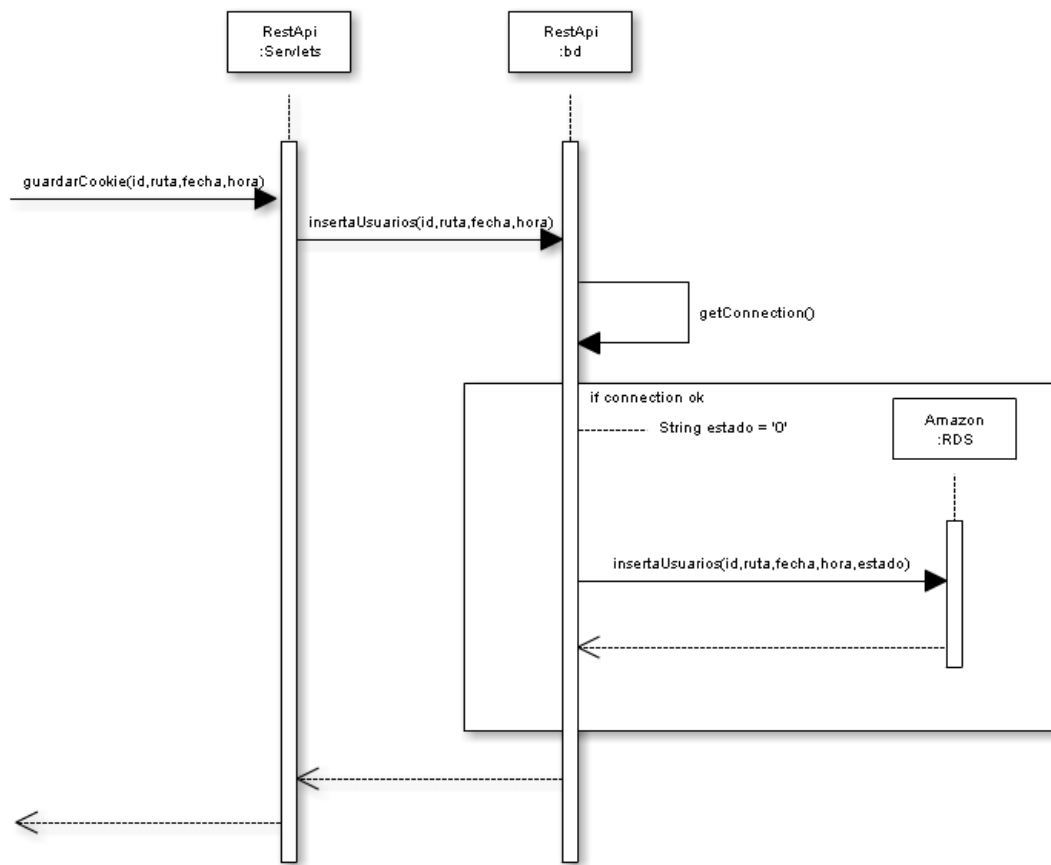
Estas opciones pueden ser seleccionadas por varios usuarios.

<sup>17</sup> Lenguaje Unificado de Modelado

### 14.3.5 Diagrama secuencia distribuir información

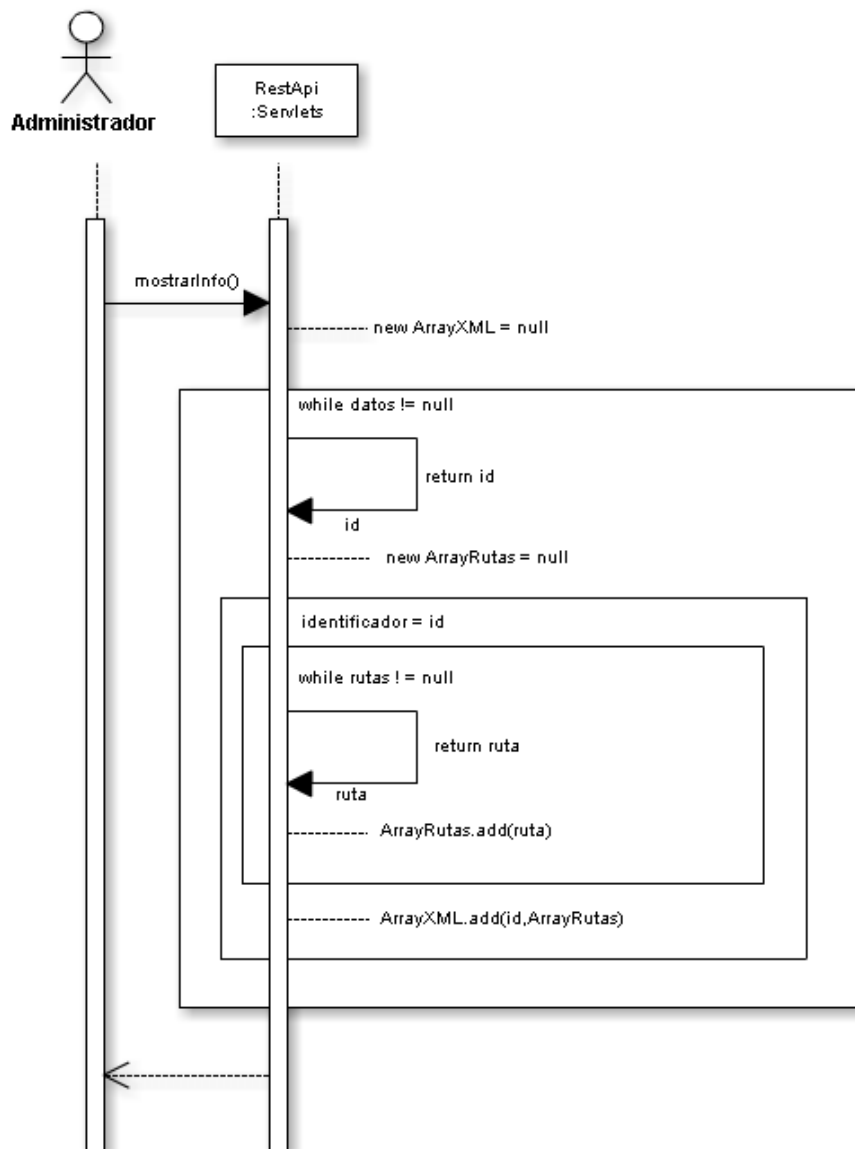


**Figura 8 : Diagrama de secuencia caso de uso distribuir información**



**Figura 9 : Diagrama de secuencia caso de uso distribuir información**

### 14.3.6 Diagrama secuencia mostrar información



**Figura 10 : Diagrama de secuencia caso de uso mostrar información**



## 15 Diseño final

En este apartado se muestra el diseño final del sistema creado en este proyecto, cumpliendo con los objetivos principales marcados en él.

### 15.1 Diseño portal web

A continuación mostraremos el diseño final del portal web.

#### 15.1.1 Página principal

En este apartado se muestra el diseño de la página principal del portal web, donde disponemos de un menú principal con los botones vehículo, partes implicadas e información, y el submenú correspondiente al botón vehículo, con los botones moto y coche. En ella se muestra que tipo de información nos ofrecerán las opciones coche y moto.



**Figura 11 : Diseño final página principal**

### 15.1.2 Página secundaria Coche

En este apartado se muestra el diseño de la página secundaria Coche del portal web, en ella podemos ver que el menú se mantiene respecto a la inicial y muestra información sobre diferentes tipos de coches con sus respectivas características.



**everis**  
an NTT DATA Company

Vehículo | Partes implicadas | Información

Coche | Moto

**Aston Martin Rapide**

**Información**

El Aston Martin Rapide es la primera berlina de cinco puertas de la marca británica, pero con un inconfundible aspecto coupé que mide 5,01 metros y que, por tanto, rivaliza directamente con el Porsche Panamera. El habitáculo ofrece espacio para cuatro pasajeros, aunque la segunda fila de asientos es bastante estrecha y sus asientos solo son recomendables para niños. Monta un motor V12 de 476 CV, que es el mismo que lleva el Aston Martin DB9 y va unido a un cambio automático transaxle (situado en la parte posterior del vehículo) de seis velocidades que en la marca denominan 'Touchtronic II'. De esta manera, se logra un reparto de pesos de 51/49. En cualquier caso el propulsor debe de arrastrar los 2.065 kilos que pesa este modelo, lo que explica que para pasar de 0 a 100 km/h necesite 5,3 segundos, una cifra más elevada que la del Panamera.

**Características**

- **Combustible:** Gasolina
- **Potencia:** 477 CV
- **Puertas:** 5
- **CO<sub>2</sub>:** 355 g/Km
- **Maletero:** 317 L

**PRECIO: 228.781 Euros**

Figura 12 : Diseño final página secundaria Coche

### 15.1.3 Página secundaria Moto

En este apartado se muestra el diseño de la página secundaria Moto del portal web, en ella podemos ver que el menú se mantiene respecto a la inicial y muestra información sobre diferentes tipos de motos con sus respectivas características.



**everis**  
an NTT DATA Company

Vehículo | Partes implicadas | Información

Coche | Moto

**Suzuki Chopper GZ 150**

**Información**

Motor confiable y potente de 150 cm3 con transmisión de 5 velocidades que asegura estabilidad y gran desempeño para el manejo en ciudad. Sistema de refrigeración Suzuki Jet (SJCS) que realiza una refrigeración activa mediante la pulverización directa del aceite de motor en la parte posterior del pistón, lo que estabiliza la temperatura en las zonas más calientes alrededor del pistón, mejorando así la eficiencia de la combustión. Inyección electrónica al alcance de todos. La GZ150 es una de las pioneras en Colombia en ofrecer el sistema de inyección electrónica para una cilindrada de 150 C.C. a precios de productos similares de esta tecnología. Este sistema garantiza un funcionamiento perfecto a cualquier golpe de aceleración y elimina las posibles fallas que un sistema tradicional de carburación presenta en viajes donde el nivel de altura sobre el nivel del mar cambia. El sistema FI (Full Injection) optimiza el consumo de combustible al controlar por computador la mezcla de combustible que requiere el motor para entregar toda su potencia. Por lo tanto esta tecnología permite ahorros del 25% o más si se compara con un motor de 150 c.c. manejado en las mismas condiciones de aceleración, tipo de combustible, peso y altura sobre el nivel del mar. La bomba de descarga del sistema de inyección electrónica (DCP- FI) permite la inyección óptima de combustible utilizando el (ECM) (Módulo de control del motor). Incluso cuando hay frío en la mañana, baja presión atmosférica (bajo nivel de oxígeno) a gran altitud, lo que mejora la capacidad de arranque, la respuesta del acelerador, la economía de combustible y respeto del medio ambiente. La rueda delantera utiliza una pinza de 2 -pod y de frenos de disco de 275 mm, consiguiendo una sensación de calidad y alto desempeño en el frenado. Estilo sólido, lujoso que hereda de la serie Boulevard de grandes cruiser de Suzuki. Asiento remachado, silenciador slash-cut, rines de aleación, suspensión trasera y detalles cromados.

**Características**

- **Cilindrada:** 150 cc
- **Potencia:** 20,5 CV
- **Transmisión:** 5 velocidades
- **Peso:** 137 Kg

**PRECIO: 7830 Euros**

Figura 13 : Diseño final página secundaria Moto

#### 15.1.4 Página secundaria Partes implicadas

En este apartado se muestra el diseño de la página secundaria Partes implicadas del portal web, en ella podemos ver que el menú principal se mantiene respecto a la inicial, mientras que el submenú ofrece las opciones UPC y Everis mediante sus respectivos botones. En ella se muestra que tipo de información nos ofrecerán las opciones UPC y Everis.



Figura 14 : Diseño final página secundaria Partes implicadas

#### 15.1.5 Página secundaria UPC

En este apartado se muestra el diseño de la página secundaria UPC del portal web, en ella podemos ver que el menú se mantiene respecto al de partes implicadas y muestra información sobre la Universidad Politécnica de Cataluña(UPC).

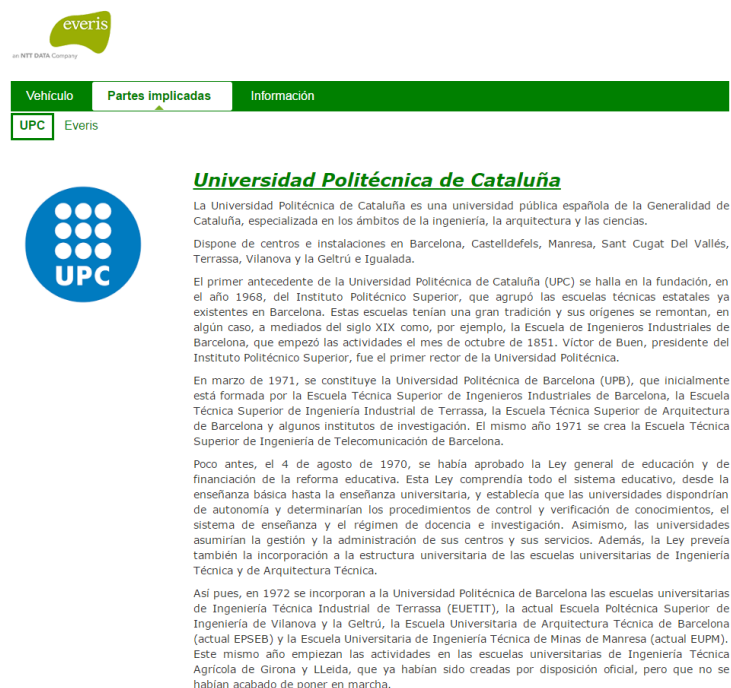


Figura 15 : Diseño final página secundaria UPC

### 15.1.6 Página secundaria Everis

En este apartado se muestra el diseño de la página secundaria Everis del portal web, en ella podemos ver que el menú se mantiene respecto al de partes implicadas y muestra información sobre la empresa Everis.



Figura 16 : Diseño final página secundaria Everis

### 15.1.7 Página secundaria Información

En este apartado se muestra el diseño de la página secundaria Información del portal web, en ella podemos ver que el menú principal se mantiene respecto a la inicial, mientras que el submenú ofrece la opción Datos mediante su respectivo botón. En ella se muestra que tipo de información nos ofrecerá las opción Datos.



Figura 17 : Diseño final página secundaria Información

### 15.1.8 Página secundaria Datos

En este apartado se muestra el diseño de la página secundaria Datos del portal web, en ella podemos ver que el menú se mantiene respecto al de información y muestra información general sobre el proyecto.



Figura 18 : Diseño final página secundaria Datos

## 15.2 Diseño Rest API

A continuación mostramos el diseño final de los casos de uso de la Rest API.

### 15.2.1 Mostrar información

En este apartado podemos ver el diseño final del método GET de nuestra Rest API, donde mostrara la información obtenida de la interacción de los usuarios por el portal web, en concreto identificador y rutas visitadas, en formato XML.

Este fichero XML no parece tener ninguna información de estilo asociada. Se muestra debajo el árbol del documento.

```
-<Datos>
  ID 0.0463747615697844841465749462506 : [http://localhost:8080/web/portal-agentes/vehiculo/moto, http://localhost:8080/web/portal-agentes/partes-implicadas/upc, http://localhost:8080/web/portal-agentes/informacion/datos]; ID 0.262049663411448641465749429987 : [http://localhost:8080/web/portal-agentes/vehiculo/coche, http://localhost:8080/web/portal-agentes/vehiculo/moto, http://localhost:8080/web/portal-agentes/informacion/datos, http://localhost:8080/web/portal-agentes/partes-implicadas/everis, http://localhost:8080/web/portal-agentes/partes-implicadas/upc]
</Datos>
```

Figura 19 : Diseño final mostrar información en XML

### **15.2.2 Almacenar datos para analizarlos**

En este apartado podemos ver el diseño final de la opción donde el administrador podrá seleccionar guardar en Amazon S3 aquellos datos de la base de datos SQL de Amazon RDS que aún no hayan sido seleccionados para ser analizados en el sistema experto. En ella podemos ver un texto explicativo de la funcionalidad de la misma así como el botón Guardar que ejecutara esta funcionalidad.



#### **Guardar datos**

Selecciona **Guardar** para almacenar la información que no ha sido guardada previamente de la base de datos



***Figura 20 : Diseño final página almacenar datos para analizarlos***

## 16 Arquitectura de Liferay

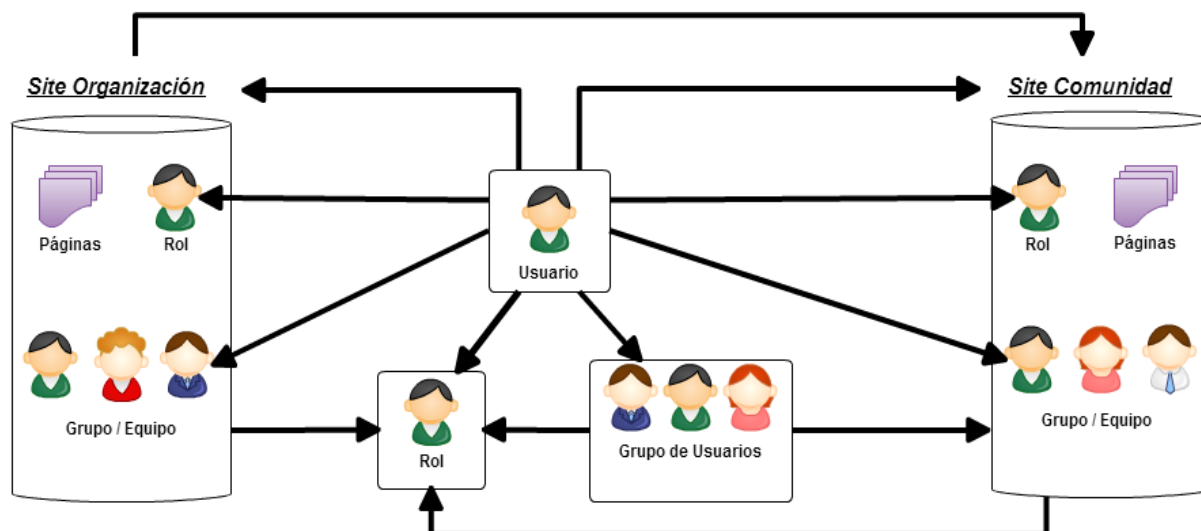
Liferay es uno de los principales y más importantes gestores de portales que hay en el mercado, debido a que es muy completo y que sigue un diseño Modelo Vista Controlador (MVC), que permite la creación y gestión de diferentes funcionalidades como páginas, usuarios, grupos y roles.

A continuación explicaremos más en detalle la arquitectura y estructura base de Liferay.

### 16.1 Instancias

Liferay está compuesto de diferentes instancias, las fundamentales y básicas son los sites(sitios web), las paginas, los usuarios, los roles, los grupos y las organizaciones.

La interacción entre estas sería como muestra la siguiente imagen.



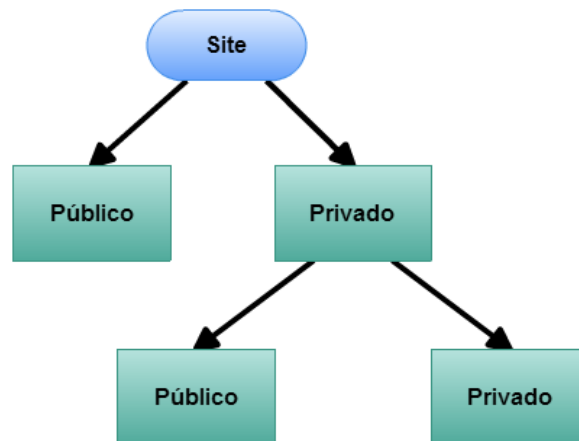
*Figura 21 : Diagrama de instancias de Liferay*

#### 16.1.1 Sites

La característica principal de un site es que está compuesto por varias páginas. Además disponemos de dos tipos diferentes, los sites de organización, los cuales nos permiten aplicar jerarquías sobre los usuarios y sites, de esta manera delegamos permisos de administración sobre ellos, y los sites de comunidad que son como los de organización con la diferencia de que estos no utilizan jerarquías y por consecuencia no delega permisos de administración.

Otra característica importante de un site es que pueden ser públicos o privados. Los públicos están accesibles a cualquier usuario del portal mientras que los privados solo

podrán acceder usuarios de ese site. Dentro de estos últimos también tenemos los que son públicos, o sea, todos los usuarios que puedan acceder a ese site lo podrán hacer desde el panel de control, o por el contrario los privados, los cuales los usuarios podrán acceder solo mediante la dirección específica de ese site y no por el panel de control.



*Figura 22 : Diagrama de sites*

### **16.1.2 Usuarios**

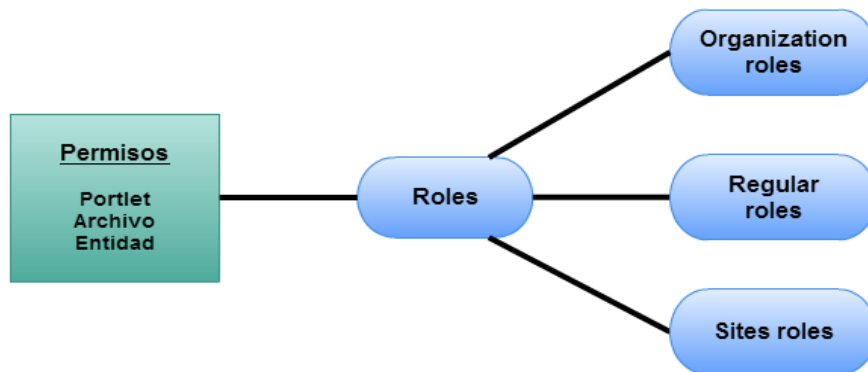
Los usuarios, como muestra la imagen inicial de más arriba (Figura 21), pueden pertenecer a diferentes sites, así como pertenecer a distintos tipos de usuarios o estar asignado con varios roles.

### **16.1.3 Roles**

Los roles permiten a Liferay gestionar los diferentes permisos sobre los recursos de que dispone como por ejemplo los portlets, los archivos y las entidades.

Liferay dispone de 3 tipos de roles, los Organization roles que indican los permisos sobre las organizaciones, los Regular roles que se encargan de indicar los permisos genéricos sobre el portal, y por último los Sites roles que indican los permisos sobre los sites.

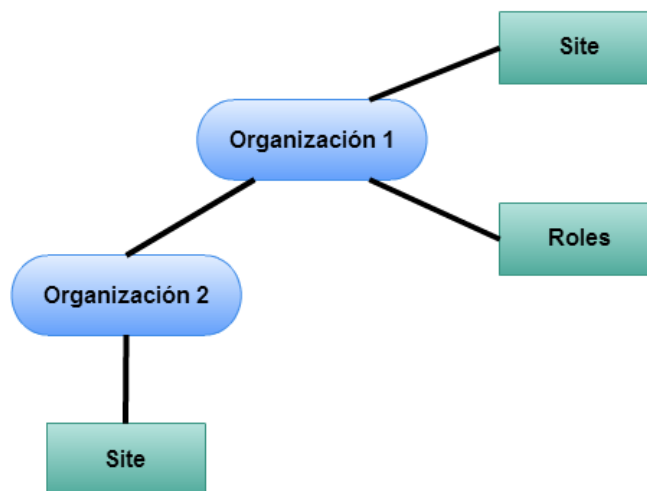




*Figura 23 : Diagrama de roles*

#### 16.1.4 Jerarquía

La jerarquía es el orden que establecemos a los usuarios dentro de una organización, dándoles a estos diferentes asignaciones a roles y sites.

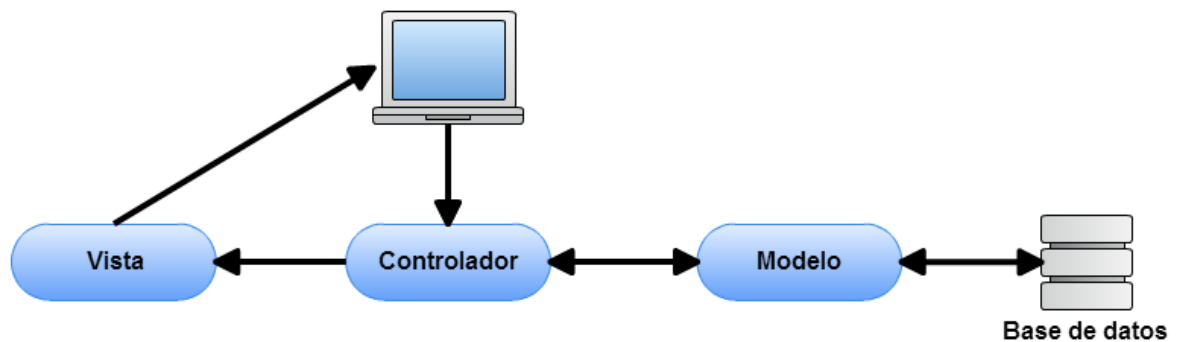


*Figura 24 : Diagrama de jerarquía*

## 16.2 Arquitectura

La arquitectura que utiliza Liferay es una arquitectura en tres capas que sigue el patrón de diseño Modelo Vista Controlador (MVC). Este patrón tiene como idea fundamental separar la presentación de la lógica de negocio.

- **Modelo:** En el modelo es donde se definen las clases de la información del mundo real.  
Esta capa se encuentra por encima de la base de datos y se encarga de realizar la gestión de estos.
- **Vista:** La vista es la capa que se encargara de toda la representación visual de la información del sistema.
- **Controlador:** El controlador es la capa que hace de intermediaria y comunica a las capas de modelo y de vista. Esta capa se encarga de gestionar todas aquellas peticiones que genere el usuario, esta recibe la petición del usuario y se la comunica a la capa de modelo que se encargara de gestionarla, una vez acabe de gestionarla la capa del controlador se encargara nuevamente de enviar la respuesta a la vista.



*Figura 25 : Diagrama Modelo-Vista-Controlador*

La estructura de Liferay será de la siguiente manera:

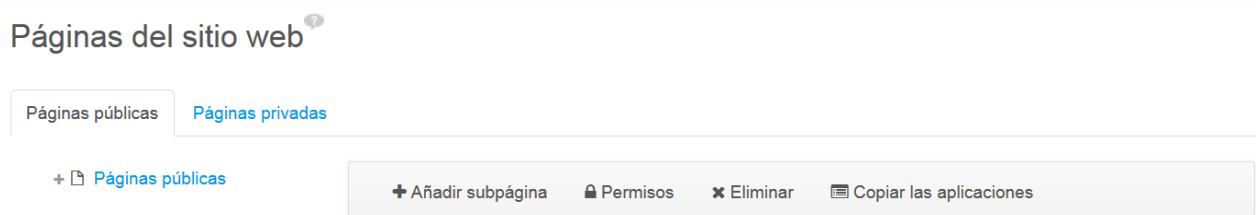
- **Persistencia:** Esta capa seria la equivalente a la capa de modelo. Su función es comunicarse con la base de datos y permitir así la persistencia de los datos del sistema.
- **Frontend:** Esta capa correspondería a la Vista y es la que comunica directamente con el usuario. En ella hacemos toda la representación visual del sistema mediante códigos HTML y CSS.
- **Servicios:** Esta capa correspondería al Controlador, su función es comunicar el frontend con la base de datos.

## 16.3 Portlet

Para la creación de nuestro portal web, hemos creado y utilizado portlets con tal de que el portal dispusiera de la información y el contenido que deseábamos.

Para administrar estos portlets Liferay nos ofrece paneles de control desde el cual gestionarlos y que explicaremos a continuación.

Desde el panel de control de páginas del sitio web, podremos gestionar si nuestros portlets son públicos o privados, así como darle diferentes permisos de usuario en función de quien queramos que lo vea. Desde este panel también podremos gestionar la eliminación de los mismos.



**Figura 26 : Panel de control páginas del sitio web**

El siguiente panel de control muestra la información interna de un portlet, como su identificador, permisos, estructura e idioma.



**Figura 27 : Panel de control interno portlet**

# 17 Implementación

A continuación se mostraran las tecnologías utilizadas en este proyecto así como ejemplos de las implementaciones más importantes del mismo.

## 17.1 Tecnologías

En este apartado listaremos las tecnologías utilizadas para el desarrollo del proyecto.

- **Liferay Developer Studio:** Programa que se utiliza para el desarrollo del proyecto en Liferay. Este utiliza Eclipse y está basado en el lenguaje Java. Los pluguins que utiliza este proyecto son liferay-pluguins-sdk-6.2-ee-sp11.
- **Apache Tomcat 7.0:** Apache es un servidor web con soporte para JSPs y servlets y que está basado en el lenguaje Java.
- **Maven:** Herramienta de software que permite gestionar y construir proyectos en Java.
- **Java:** Lenguaje principal sobre el que se ha construido el proyecto en Liferay. Este es un lenguaje derivado de C y C++ orientado a objetos.
- **JavaScript:** Es un lenguaje de programación orientado a objetos, que nos permite la interacción con los portales webs ejecutándolo en el navegador del cliente, haciendo de esta manera la web dinámica. Este lenguaje ofrece al usuario la posibilidad de interactuar con la interfaz sin necesidad de hacer nuevas peticiones al servidor.
- **JSP:** JavaServer Page es un lenguaje basado en Java similar a PHP, y que permite la creación de páginas web dinámicas.
- **AJAX:** Técnica de desarrollo web para crear aplicaciones interactivas.
- **HTML:** Lenguaje de marcas de hipertexto para la elaboración de páginas web. Define el contenido de estas.
- **XML:** Lenguaje de marcas utilizado para el almacenamiento de datos de una forma legible.
- **CSS:** Lenguaje que define y crea la presentación de documentos estructurados como HTML o XML.

- **MySQL:** Sistema de gestión de base de datos relacional de código abierto y multiusuario.
- **MySQL Workbench 6.2 CE:** Programa que permite manejar y gestionar bases de datos del tipo MySQL.
- **Toad MySQL:** Programa que permite manejar y gestionar bases de datos del tipo MySQL.
- **Amazon RDS MySQL:** Base de datos de Amazon que permite tratar y gestionar datos del tipo MySQL.
- **Amazon IAM:** Permite crear usuarios de Amazon, los cuales podremos darles permisos de accesos en las diferentes funcionalidades de Amazon.
- **Amazon S3:** Sistema de almacenamiento de archivos de Amazon.
- **Amazon Machine Learning:** Sistema experto de Amazon que ofrece la capacidad de analizar e interpretar los datos que previamente le hayamos pasado. Este ofrece la posibilidad de calibrar y configurar el mismo para obtener la mayor precisión y fiabilidad posible en el análisis de estos.

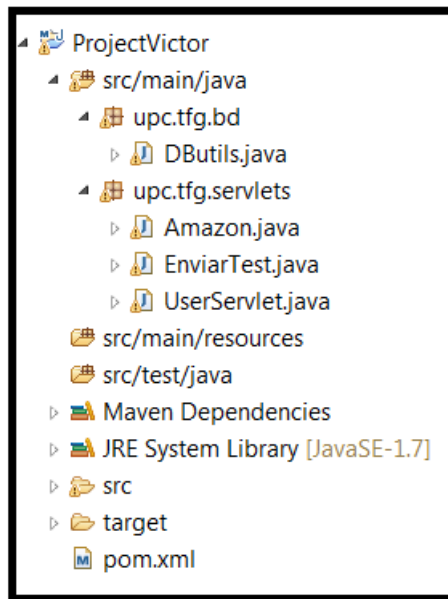
## 17.2 Obtener información interacción web

Para obtener la información de la interacción de los usuarios por la web, utilizamos un javascript en el cual hemos generado varias funciones que obtienen o crean datos como el identificador de la sesión que le damos al usuario que interactúa, la ruta por la cual ha navegado, así como la fecha y hora de la misma. La función más importante y relevante en el mismo es la que comunicara esta información a la Rest API. Esta es la función guardarCookie, que mediante Ajax comunica al método Post de nuestra Rest API indicada en el parámetro url, los datos pasados como parámetros en la misma.

```
function guardarCookie(id,name,fechaActual,horaActual){
    var dataString = {
        "id"      : id,
        "name"    : name,
        "fechaActual" : fechaActual,
        "horaActual" : horaActual,
    };
    $.ajax({
        url : 'http://localhost:8080/ProjectVictor/UserServlet',
        data : dataString,
        type : 'POST',
        dataType : 'xml',
    })
}
```

*Figura 28 : Function guardarCookie*

## 17.3 Rest Api



**Figura 29 : Esquema Rest Api**

Uno de los métodos más importantes de la Rest API es el método doPost, este recibe la información de la interacción del usuario por la web explicada en el punto anterior mediante llamadas HTTP , esta información recibida la utilizamos para que el método doGet pueda tratarla y mostrarla en formato XML, y también comunicarla a nuestra base de datos.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    accion(request, response);
}

private void accion(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String parametro1 = request.getParameter("id");
    String parametro2 = request.getParameter("name");
    String parametro3 = request.getParameter("fechaActual");
    String parametro4 = request.getParameter("horaActual");
    dbutils.insertaUsuarios(parametro1, parametro2, parametro3, parametro4);
    param1 = param1 + parametro2 + "=" + parametro1 + ";";
}
```

**Figura 30 : Método POST Rest Api**

Para hacer la inserción de estos datos en la base de datos, hemos creado el método insertaUsuarios, este hará la conexión mediante el método getConnection a la base de datos y le enviara una llamada del tipo SQL para que la ejecute con los parámetros que deseamos que trate.

```

public boolean insertaUsuarios(String idUser, String ruta, String fechaActual, String horaActual) {
    String insertUser = "Insert into usuarios4 values(?,?,?,?)";
    CallableStatement cs = null;
    try {
        Connection conn = getConnection();
        cs = conn.prepareCall(insertUser);
        cs.setString(1, idUser);
        cs.setString(2, ruta);
        cs.setString(3, fechaActual);
        cs.setString(4, horaActual);
        cs.setString(5, "0");
        boolean result = cs.execute();
        return result;
    }
    catch (SQLException e) {
        e.printStackTrace();
        return false;
    }
    finally {
        closeConnection(cs);
    }
}

```

**Figura 31 : Inserción de datos en la base de datos**

El método getConnection recibirá la petición de conexión y cargará la información de conexión a la base de datos como la url, username y password que almacenamos en un fichero Properties.

```

private Connection getConnection() throws SQLException {
    try {
        System.out.println(DBUtils.class.forName("com.mysql.jdbc.Driver"));
    }
    catch (ClassNotFoundException e) {
    }
    Properties info = new Properties();
    info.put("user", user);
    info.put("password", password);
    Connection conn = DriverManager.getConnection(urlBD, info);
    return conn;
}

```

**Figura 32 : Conexión a la base de datos**

Otro punto importante sería la conexión con Amazon S3 para poder almacenar la creación del fichero de datos aun no tratados por el sistema experto y que se almacenara en AmazonS3 para su posterior análisis. El método connexionAmazonS3 nos permite establecer la conexión con Amazon S3 con la información que contiene el fichero Properties, como el identificador, password o nombre del path donde se almacenara. La llamada putObject es la que cargara el fichero de datos en AmazonS3.

```
public void connexionAmazonS3(String nombreFichero) throws IOException{
    try {
        Properties info = cargaficheroS3();
        String keyID = info.getProperty("aws_access_key_id");
        String keyAccess = info.getProperty("aws_secret_access_key");
        String nameBucket = info.getProperty("bucketS3");

        BasicAWSCredentials awsCreds = new BasicAWSCredentials(keyID, keyAccess);
        AmazonS3 s3 = new AmazonS3Client(awsCreds);
        Region euWest1 = Region.getRegion(Regions.EU_WEST_1);
        s3.setRegion(euWest1);

        File f = new File("/") + nombreFichero;
        s3.putObject(new PutObjectRequest(nameBucket, nombreFichero, f));
    }
    catch (AmazonServiceException ase) {
        System.out.println("Caught an AmazonServiceException, which means your request made it "
            + "to Amazon S3, but was rejected with an error response for some reason.");
        System.out.println("Error Message: " + ase.getMessage());
        System.out.println("HTTP Status Code: " + ase.getStatusCode());
        System.out.println("AWS Error Code: " + ase.getErrorCode());
        System.out.println("Error Type: " + ase.getErrorType());
        System.out.println("Request ID: " + ase.getRequestId());
    }
    catch (AmazonClientException ace) {
        System.out.println("Caught an AmazonClientException, which means the client encountered "
            + "a serious internal problem while trying to communicate with S3, "
            + "such as not being able to access the network.");
        System.out.println("Error Message: " + ace.getMessage());
    }
}
}
```

**Figura 33 : Conexión AmazonS3**

Otro aspecto importante es la necesidad de disponer de dependencias o librerías. En el fichero pom.xml se han introducido todas aquellas necesarias, por ejemplo una de las más importantes sería la de MySQL.

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.38</version>
</dependency>
```

**Figura 34 : Dependencia MySQL**



## 17.4 Análisis e interpretación de datos

Tras completar la parte del proyecto dedicada a la creación de la REST API, ya se puede pasar a la fase de análisis e interpretación de datos, en la cual se configurara y calibrara diferentes sistemas de Amazon, concretamente Amazon RDS, Amazon IAM, Amazon S3 y Amazon Machine Learning, así como programas que permiten el manejo y la gestión de bases de datos SQL como Toad MySQL[8] o MySQL Workbench[9], que son los que nos permitirán cumplir con uno de los objetivos principales del proyecto que era la interpretación de los datos.

La decisión de utilizar los sistemas de Amazon, ha sido debido a que después del estudio previo realizado nos ofrecían lo que buscábamos para el proyecto y a la empresa en la cual realizo este le resulta beneficioso ya que utilizan con mucha frecuencia y en muchos de sus proyectos sistemas proporcionados por Amazon ya que estos se alinean correctamente con las tecnologías de desarrollo de esta, ya que les ofrece principalmente una alta escalabilidad, flexibilidad, seguridad y servicio técnico.

A continuación enumeraremos los pasos a realizar en estos sistemas para poder hacer la interpretación de datos.

### 17.4.1 Requisitos

El requisito principal que tendremos que cumplir es tener una cuenta en Amazon Web Service para poder utilizar sus sistemas.

1. Acceder a Amazon a través de la siguiente URL : <https://aws.amazon.com/es/>
2. Acceder a la página donde nos logeamos clicando el botón de **Inscribirse**.
3. Acceder a la cuenta de Amazon introduciendo nuestro e-mail y escogiendo la opción **"Soy un usuario registrado y mi contraseña es:"** donde se introducirá la contraseña de la cuenta correspondiente a la dirección de e-mail indicada previamente.

### 17.4.2 Amazon RDS

Una vez hecho el login en nuestra cuenta de Amazon comenzamos a trabajar en la configuración de Amazon RDS.

1. Seleccionar del panel inicial Amazon **RDS**.
2. En el menú izquierdo, presionar el link **Instances**.
3. Para crear una nueva instancia de base de datos clicamos el botón azul de la parte superior **Launch DB Instance**.
4. De la lista mostrada, escogeremos la imagen del tipo de base de datos que queremos, en este caso **MySQL**, y clicaremos el botón azul **Select** que se habilitara al seleccionar la imagen.

5. Seleccionamos el tipo de datos MySQL marcando la casilla **Dev/Test**, para desarrolladores y testing. Acto seguido clicamos el botón azul de la parte inferior derecha **Next Step**.

6. Procedemos a configurar la instancia seleccionando.

- Tipo de licencia: **general-public-license**
- Tipo de instancia: **t1.micro** (1 CPU, 0,613 GiB de memoria RAM)
- Capacidad de almacenamiento: **5 GB**
- Insertamos nombre Identificador de la Instancia
- Insertamos Username
- Insertamos Password

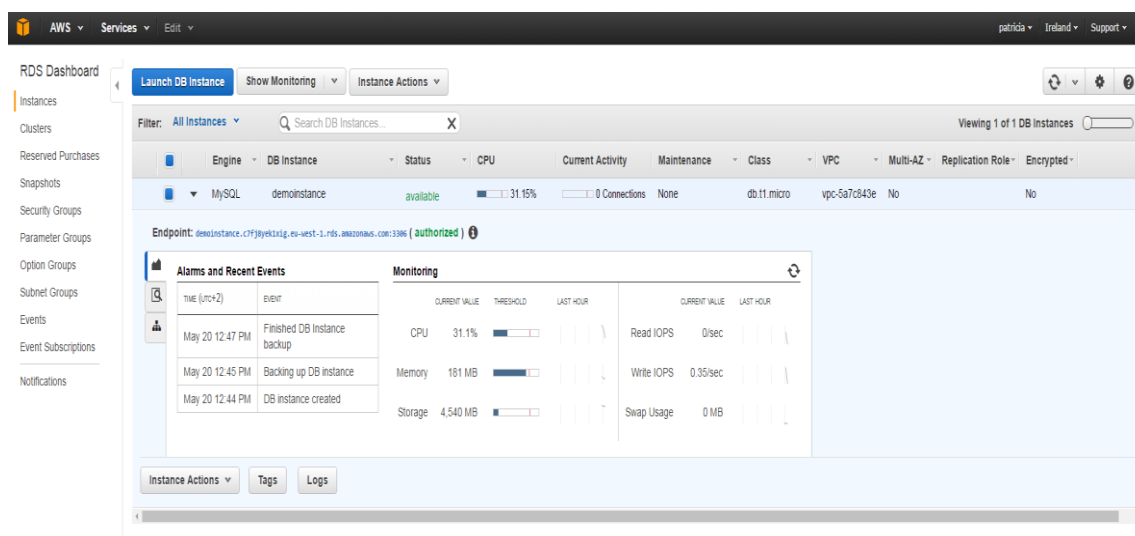
Acto seguido clicamos el botón azul de la parte inferior derecha **Next Step**.

7. En la página siguiente dejamos los parámetros de seguridad y de backups con los valores por defecto. Introducimos un nombre a nuestra base de datos.

8. Una vez revisado que todos los parámetros introducidos son correctos procedemos a crear la instancia clicando el botón azul de la parte inferior derecha **Launch DB Instance**.

9. A continuación se mostrara el mensaje en verde **“Your DB Instances is being created”** que nos indicara que nuestra instancia de base de datos se ha creado correctamente. Clicamos el botón azul de la parte inferior derecha **View Your DB Instances** para visualizarla.

10. A continuación se mostrara la lista de instancias creadas. Marcamos la casilla de la instancia que acabamos de crear y se desplegara el panel de control desde el cual controlarla.

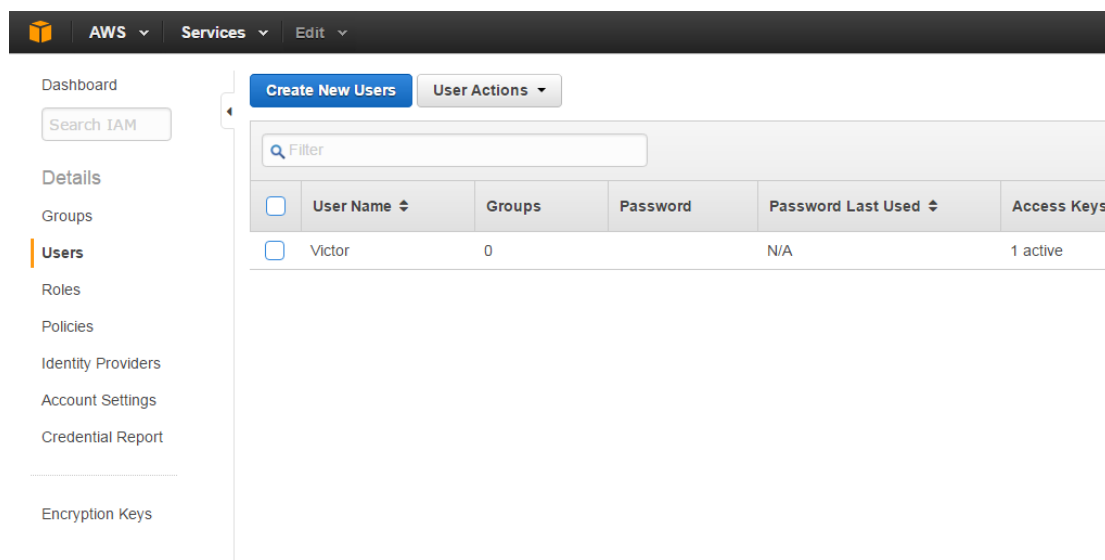


**Figura 35 : Panel de control base de datos Amazon RDS MySQL**

### 17.4.3 Amazon IAM

Una vez hecho el login en nuestra cuenta de Amazon comenzamos a trabajar en la creación de usuarios.

1. Seleccionamos del panel inicial Amazon **IAM**.
2. Seleccionamos del menú de la parte izquierda la opción **Users**.
3. Clicamos sobre el botón azul de la parte superior **Create New User**.
4. A continuación introducimos el nombre del usuario que creamos en **Enter User Names** y seleccionamos el botón azul de la parte inferior izquierda **Create**.
5. Acto seguido se visualiza el usuario creado con sus claves.
6. Volvemos al menú principal de IAM, y clicamos sobre la opción **Users** del menú izquierdo para visualizar los usuarios creados. Clicando sobre ellos consultamos la información de estos.



**Figura 36 : Panel de control usuarios Amazon IAM**

### 17.4.4 Amazon S3

Una vez hecho el login en nuestra cuenta de Amazon comenzamos a trabajar en la configuración de Amazon S3.

1. Seleccionar del panel inicial Amazon **S3**.
2. Creamos un nuevo lugar de almacenamiento, clicando sobre el botón azul **Create Bucket**.
3. Introducimos el **nombre** del bucket, **la región** y seleccionamos el botón azul **Create**.

4. A continuación se mostrara la lista con todos los lugares de almacenamientos creados.



*Figura 37 : Panel de Buckets Amazon S3*

5. Clicamos sobre el **nombre** del bucket creado para consultar que contiene.

6. A continuación clicamos sobre el botón de la parte superior derecha **Properties** para configurar este bucket.

7. Desplegamos la opción **Permissions** clicando sobre él y clicamos el botón **Add more permissions** que aparecerá, para dar permisos de usuario sobre este bucket.

8. Acto seguido aparecerá un desplegable **Grantee** del cual seleccionamos la opción **Any Authenticated AWS User** y marcamos las opciones **List, Upload/Delete, View Permissions y Edit Permissions** que se muestran a su lado.

9. Seleccionamos el botón azul **Save** para guardar los nuevos datos de permisos sobre el bucket.

#### **17.4.5 Amazon Machine Learning**

Una vez tengamos configurada y operativa nuestra base de datos de Amazon RDS, así como Amazon IAM y Amazon S3, ya podemos comenzar a trabajar en la configuración de Amazon Machine Learning para la interpretación y análisis de los datos. Tendremos que hacer login para poder trabajar con este sistema.

1. Seleccionar del panel inicial **Amazon Machine Learning**.

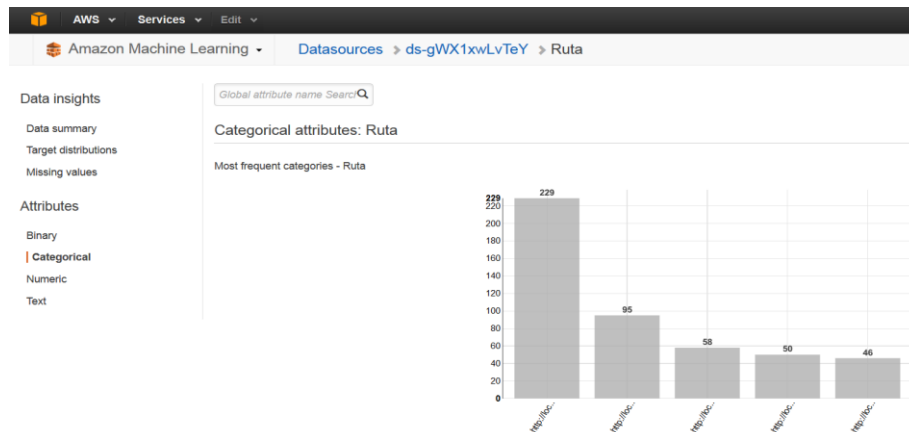
2. Comenzamos a utilizar este sistema clicando sobre el botón azul **Launch**.

3. Entramos dentro del panel de selección de datos a introducir en el sistema, marcamos la opción **S3** que es donde están almacenados los datos que hemos ido obteniendo de la base de datos SQL de Amazon RDS. A continuación, introducimos en sus campos correspondientes el **path** de **S3** donde están ubicados estos datos y el nombre que le daremos. Por ultimo clicamos el botón azul de la parte inferior derecha **Verify**.

4. Acto seguido se mostrara un aviso de solicitud de permisos de Amazon Machine Learning sobre Amazon S3. Clicamos el botón azul de la parte inferior derecha **Yes**.
5. A continuación se mostrara un aviso de que se ha validado correctamente la operación. Con el mensaje **“The validation is successful. To go to the next step, choose Continue”** y los parámetros principales de esta. Clicamos el botón azul de la parte inferior derecha **Continue**.
6. En la página siguiente se mostrara los datos que hemos cargado de AmazonS3 para Amazon Machine Learning, sus atributos (en este caso son Identificador, Ruta, Fecha y Hora), tipo de datos e información que contiene.
7. Procedemos a comprobar que son correctos y para cada atributo seleccionamos que tipo de datos son, marcamos en la opción **target** de la tabla el atributo que utilizamos para el modelo de aprendizaje. Clicamos el botón azul de la parte inferior derecha **Continue**.
8. Acto seguido se mostrara el mensaje **Does your data contain an identifier?**, en el que marcamos la opción **No**. Clicamos el botón azul de la parte inferior derecha **Review**.
9. Se mostrara un listado con todas las opciones y parámetros que se han configurado hasta el momento, y nos ofrece la opción de modificarlos con el botón de la parte derecha **Edit**. Dejamos todo tal y como esta y clicamos el botón azul de la parte inferior derecha **Continue**.
10. A continuación comprobamos que los parámetros del modelo de aprendizaje a generar son correctos, y dejamos todo tal y como esta con la opción Default marcada. Clicamos el botón azul de la parte inferior derecha **Review**.
11. En la página siguiente saldrá un listado con todas las opciones y parámetros del modelo, y nos ofrece la opción de modificarlos con el botón de la parte derecha **Edit**. Dejamos todo tal y como esta y clicamos el botón azul de la parte inferior derecha **Finish**.
12. Acto seguido se mostrara información sobre el modelo, como por ejemplo su estado que inicialmente será **Pending**, pasara a In Progress y posteriormente pasara a ser **Complete**.
13. Clicamos en la parte superior izquierda sobre **Amazon Machine Learning** y desplegamos el menú, seleccionamos **Dashboard** para ir al panel de control donde estará la información.
14. Podemos desplegar cualquier opción de la lista marcando sobre el **nombre** y veremos su información. Seleccionamos una de las opciones.

15. Se mostrara toda la información sobre el mismo. Clicamos sobre el menú izquierdo una de las opciones de **Attributes** del tipo utilizado. Se desplegara la información sobre todos los atributos de ese tipo que hayamos tratado.

16. Clicamos encima de la opción **Preview** del atributo, para ver la información en formato gráfico.



**Figura 38 : Opción Preview Amazon Machine Learning**

17. Clicamos en la parte superior izquierda sobre **Amazon Machine Learning** desplegamos el menú, seleccionamos **Dashboard** para ir al panel de control.

18. Seleccionamos **ML model**.

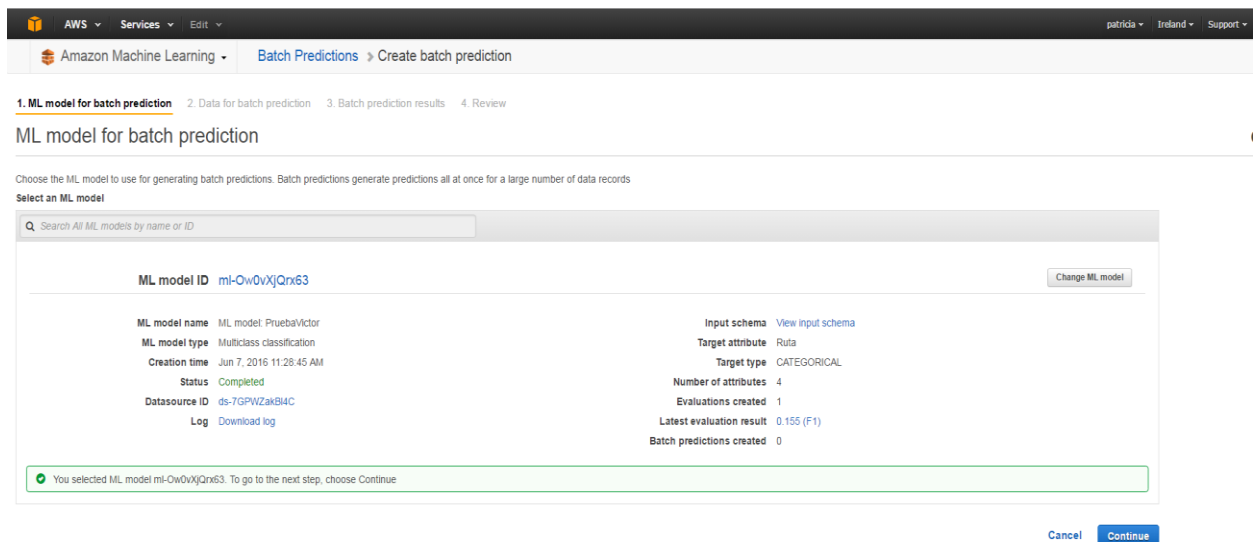
19. Desplegamos del menú izquierdo **Evaluation** y seleccionamos **Summary**. Veremos un mensaje que nos dirá si el modelo de aprendizaje es correcto o no.

19. Desplegamos del menú izquierdo **Evaluation** y seleccionamos **Explore Performance** para ver información del modelo de aprendizaje.

20. Clicamos en la parte superior izquierda sobre **Amazon Machine Learning** desplegamos el menú, seleccionamos **Batch Predictions** para hacer una predicción.

21. A continuación clicamos el botón azul de la parte superior izquierda **Create new batch Prediction**.

22. Seleccionamos de la lista que se muestra, el **modelo** recientemente creado. Saldrá un mensaje de confirmación en la parte inferior. Clicamos el botón azul de la parte inferior derecha **Continue**.



**Figura 39 : Selección y confirmación modelo de datos**

23. A continuación marcamos la opción **“My data is in S3, and I need to create a datasource”** para indicarle de donde vienen los datos. Clicamos el botón azul de la parte inferior derecha **Continue**.

24. Introducimos el **nombre** y el **path** de donde proceden los datos en sus respectivos campos. Marcamos **Yes** a la pregunta **“Does the first line in your CSV contain the column names?”**. Clicamos el botón azul de la parte inferior derecha **Verify**.

25. Acto seguido se muestra un mensaje de que se ha validado correctamente. Clicamos el botón azul de la parte inferior derecha **Continue**.

26. Introducimos en el campo **S3 destination**, el **path** de Amazon S3 donde se guardara el resultado de la predicción y dejamos el campo de nombre que viene por defecto. Clicamos el botón azul de la parte inferior derecha **Review**.

27. A continuación se mostrara toda la información introducida, con la opción de modificarlo con el botón **Edit** de la parte derecha. Dejamos todo tal y como esta y clicamos el botón de la parte inferior derecha **Finish** para efectuar la predicción.

28. Se mostrara un mensaje con la información y el estado **In progress**, cuando finaliza muestra **Complete**.



**Figura 40 : Panel estado predicción**

29. Clicamos en la parte superior izquierda sobre **Amazon Machine Learning** desplegamos el menú, seleccionamos **Batch Predictions** para ver las predicciones.

30. Marcamos nuestra predicción de la lista para ver sus información.

31. Desplegamos de la parte superior del menú de Amazon la opción **Services** y seleccionamos **S3**.

32. A continuación vamos al **path** indicado previamente donde se guardaban los resultados.

33. Abrimos la carpeta **batch-prediction** y dentro la carpeta **result**.



**Figura 41 : Panel de almacenamiento resultados en AmazonS3**

34. Clicamos sobre el fichero **.gzip** que contiene los resultados para descargarlo.

35. Abrimos fichero **.gzip**.



#### **17.4.6 Toad MySQL**

Toad MySQL es una herramienta que servirá para poder gestionar la base de datos MySQL de Amazon RDS y que se configura de la siguiente manera.

1. Se crea una nueva conexión
2. Se añade un nombre
3. Se introduce el **Hostname**, en este caso será el proporcionado por Amazon RDS.
4. Se introduce el **Puerto**, es este caso será el proporcionado por Amazon RDS.
5. Se introduce el **Username**.
6. Se introduce el **Password**.
7. Confirmamos la conexión dando **OK**.
8. Iniciamos la conexión.

Una vez establecida la conexión podremos gestionar a través de esta herramienta la base de datos MySQL de Amazon RDS.

#### **17.4.7 MySQL Workbench**

MySql Workbench es una herramienta que servirá para poder gestionar la base de datos MySQL de Amazon RDS y que se configura de la siguiente manera.

1. Se crea una nueva conexión
2. Se añade un nombre
3. Se introduce el **Hostname**, en este caso será el proporcionado por Amazon RDS.
4. Se introduce el **Puerto**, es este caso será el proporcionado por Amazon RDS.
5. Se introduce el **Username**.
6. Se introduce el **Password**.
7. Confirmamos la conexión dando **OK**.
8. Iniciamos la conexión.

Una vez establecida la conexión podremos gestionar a través de esta herramienta la base de datos MySQL de Amazon RDS.

## 18 Planificación, análisis de costes y metodologías

### 18.1 Modificaciones realizadas en la planificación

En la planificación inicial, se indicó que la dedicación que se haría sobre el proyecto sería de 4 horas diarias, en un horario de 9:00 a 13:00 según las horas establecidas en el convenio con la empresa. Siguiendo la planificación inicial, el proyecto estaba previsto que finalizase el 24 de Junio de 2016 incluyendo la preparación de la lectura.

Después de las dificultades encontradas en el Sprint inicial respecto a la fase de implementación, que posteriormente se llevaría a cabo en los siguientes Sprint, se tomó la decisión de ampliar en una hora más diaria la dedicación sobre el proyecto en la fase de implementación del mismo, para de esta manera poder cumplir correctamente las fechas de entrega marcadas para cada Sprint y por consiguiente la fecha de entrega marcada en la planificación inicial.

Los cambios efectuados sobre la planificación inicial después de este cambio de dedicación de horas diarias será el siguiente. Marcado en negrita indicaremos la dedicación real al proyecto, mientras que la parte no marcada será la planificada inicialmente.

Tarea	Duración	Costes
<b>Sprint Inception</b>	11/01/2016 – 24/03/2016	216 horas
	<b>11/01/2016 – 24/03/2016</b>	<b>216 horas</b>
<b>Sprint 1</b>	25/03/2016 – 19/04/2016	72 horas
	<b>25/03/2016 – 19/04/2016</b>	<b>90 horas</b>
<b>Sprint 2</b>	20/04/2016 – 13/05/2016	72 horas
	<b>20/04/2016 – 13/05/2016</b>	<b>90 horas</b>
<b>Sprint 3</b>	16/05/2016 – 08/06/2016	72 horas
	<b>16/05/2016 – 08/06/2016</b>	<b>90 horas</b>
<b>Sprint 4</b>	09/06/2016 – 24/06/2016	48 horas
	<b>09/06/2016 – 24/06/2016</b>	<b>48 horas</b>
		Total: 480 horas
		<b>Total: 534 horas</b>

*Tabla 14: Tabla de comparación costes de dedicación inicial-final*












#### 18.1.1 Sprint Inception: Investigación, planificación y puesta en marcha

En el Sprint Inception en el que lleve a cabo la investigación previa necesaria para la realización del proyecto consultando documentación, así como preparar el entorno de desarrollo y hacer el aprendizaje de las herramientas que se utilizan y hacer la documentación de gestión de proyectos (GEP), no se encuentra ninguna diferencia de la planificación inicial a la dedicación real. En este Sprint en cambio se tomó la decisión, debido a haber encontrado más dificultades de las previstas en el aprendizaje

básico de las herramientas a utilizar, de ampliar en una hora la dedicación diaria al proyecto en los siguientes Sprint dedicados a la fase de implementación del mismo, con tal de poder cumplir las fechas de entrega marcadas en la planificación inicial.

La fecha de inicio de este Sprint fue el 11/01/2016 y finalizo 24/03/2016 con un total de 216 horas de dedicación.

#### **18.1.1.1 Sprint inception final**

		Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		<input type="checkbox"/> Sprint Inception: Investigación, planificación y puesta en marcha	54d?	11/01/2016	24/03/2016		
2		<input type="checkbox"/> Investigación previa	30d?	11/01/2016	19/02/2016		
3		Definición y preparación entorno de desarrollo	20d?	11/01/2016	05/02/2016		victor
4		Aprendizaje entorno de desarrollo	10d?	08/02/2016	19/02/2016	3	victor
5		<input type="checkbox"/> Gestión del proyecto (GEP)	23d?	22/02/2016	23/03/2016	2	
6		Alcance del proyecto y contextualización	7d?	22/02/2016	01/03/2016		victor
7		Planificación temporal	3d?	02/03/2016	04/03/2016	6	victor
8		Gestión económica y sostenibilidad	3d?	07/03/2016	09/03/2016	7	victor
9		Presentación preliminar	2d?	10/03/2016	11/03/2016	8	victor
10		Condiciones especialidad	3d?	14/03/2016	16/03/2016	9	victor
11		Preparación documento final	3d?	17/03/2016	21/03/2016	10	victor
12		Presentación	2d?	22/03/2016	23/03/2016	11	victor
13		Sprint Review	1d?	24/03/2016	24/03/2016	5	victor

**Figura 42 : Tareas y dedicación Sprint Inception final**

#### **18.1.2 Sprint 1: Implementación REST API y documentación**

En el Sprint 1 que es donde se llevaba a cabo la implementación de la REST API, se cumplieron las fechas previstas en la planificación inicial, pero para ello se tuvo que tomar la decisión de aumentar en 1 hora más la dedicación diaria a este Sprint, lo cual resulto una decisión acertada ya que con esta modificación se pudo cumplir tanto con las fechas marcadas en la definición de la estructura, las diferentes implementaciones, GET, POST, así como la realización de las diferentes pruebas de testeo y la redacción de la documentación final.

La fecha de inicio de este Sprint fue el 25/03/2016 y finalizo 19/04/2016 con un total de 90 horas de dedicación.

### 18.1.2.1 Sprint 1 final

		Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		☒ Sprint Inception: Investigación, planificación y puesta en mar	54d?	11/01/2016	24/03/2016		
14		☒ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
15		Definición estructura REST API	3d?	25/03/2016	29/03/2016		victor
16		Implementación método POST	5d?	30/03/2016	05/04/2016	15	victor
17		Implementación método GET	5d?	06/04/2016	12/04/2016	16	victor
18		Testing	2d?	13/04/2016	14/04/2016	17	victor
19		Redacción memoria final	2d?	15/04/2016	18/04/2016	18	victor
20		Sprint Review	1d?	19/04/2016	19/04/2016	19	victor







**Figura 43 : Tareas y dedicación Sprint 1 final**

### 18.1.3 Sprint 2: Configuración sistema experto y documentación

En el Sprint 2 que es donde se llevaba a cabo la configuración del sistema experto que se utiliza para el análisis e interpretación de los datos que obtenemos de la interacción de los usuarios por un portal web, se cumplieron las fechas previstas en la planificación inicial, pero para ello se tuvo que tomar la decisión de aumentar en 1 hora más la dedicación diaria a este Sprint, lo cual resulto una decisión acertada ya que la definición y configuración de este sistema resulto ser más problemática de lo que inicialmente se había estimado. Con esta modificación de horas se pudo cumplir tanto con las fechas marcadas en la definición del sistema, configuración del sistema, así como la realización de las diferentes pruebas de testeo y la redacción de la documentación final.

La fecha de inicio de este Sprint fue el 20/04/2016 y finalizo 13/05/2016 con un total de 90 horas de dedicación.

#### 18.1.3.1 Sprint 2 final

		Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		☒ Sprint Inception: Investigación, planificación y puesta en mar	54d?	11/01/2016	24/03/2016		
14		☒ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
21		☒ Sprint 2: Configuración sistema experto y documentación	18d?	20/04/2016	13/05/2016	14	
22		Definición sistema experto	3d?	20/04/2016	22/04/2016		victor
23		Configuración sistema experto	10d?	25/04/2016	06/05/2016	22	victor
24		Testing	2d?	09/05/2016	10/05/2016	23	victor
25		Redacción memoria final	2d?	11/05/2016	12/05/2016	24	victor
26		Sprint Review	1d?	13/05/2016	13/05/2016	25	victor

**Figura 44 : Tareas y dedicación Sprint 2 final**

#### **18.1.4 Sprint 3: Implementación web - REST API - sistema experto y documentación**

En el Sprint 3 que es donde se llevaba a cabo la implementación de la estructura web y la comunicación de los tres elementos web-REST API-Sistema experto, se cumplieron las fechas previstas en la planificación inicial, pero para ello se tuvo que tomar la decisión de aumentar en 1 hora más la dedicación diaria a este Sprint. Con esta modificación de horas se pudo cumplir tanto con las fechas marcadas en la definición de la estructura, implementación de la estructura, así como la realización de las diferentes pruebas de testeo y la redacción de la documentación final.

La fecha de inicio de este Sprint fue el 16/05/2016 y finalizo 08/06/2016 con un total de 90 horas de dedicación.

##### **18.1.4.1 Sprint 3 final**

		Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		☒ Sprint Inception: Investigación, planificación y puesta en marcha	54d?	11/01/2016	24/03/2016		
14		☒ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
21		☒ Sprint 2: Configuración sistema experto y documentación	18d?	20/04/2016	13/05/2016	14	
27		☒ Sprint 3: Implementación web - REST API - sistema experto y documentación	18d?	16/05/2016	08/06/2016	21	
28		Definición estructura	3d?	16/05/2016	18/05/2016		victor
29		Implementación web	5d?	19/05/2016	25/05/2016	28	victor
30		implementación web - REST API - sistema experto	5d?	26/05/2016	01/06/2016	29	victor
31		Testing	2d?	02/06/2016	03/06/2016	30	victor
32		Documentación memoria final	2d?	06/06/2016	07/06/2016	31	victor
33		Sprint Review	1d?	08/06/2016	08/06/2016	32	victor

**Figura 45 : Tareas y dedicación Sprint 3 final**

#### **18.1.5 Sprint 4: Documentación final**

En el Sprint 4 que es donde se llevaba a cabo la finalización de la elaboración de la memoria final del proyecto así como la preparación de la lectura, no se ha sufrido ningún cambio ya sea positivo o negativo respecto a la planificación prevista inicialmente, cumpliendo de esta manera las fechas de entrega pautadas inicialmente. El hecho de haber ido efectuando la memoria a lo largo de todos los Sprint ha sido fundamental para que en este último Sprint no hayamos tenido ninguna desviación negativa.

La fecha de inicio de este Sprint fue el 09/05/2016 y finalizo 24/06/2016 con un total de 48 horas de dedicación.

#### 18.1.5.1 Sprint 4 final

		Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		⊕ Sprint Inception: Investigación, planificación y puesta en mar	54d?	11/01/2016	24/03/2016		
14		⊕ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
21		⊕ Sprint 2: Configuración sistema experto y documentación	18d?	20/04/2016	13/05/2016	14	
27		⊕ Sprint 3: Implementación web - REST API - sistema experto y	18d?	16/05/2016	08/06/2016	21	
34		⊖ Sprint 4: Documentación final	12d?	09/06/2016	24/06/2016	27	
35		Redacción de la memoria	5d?	09/06/2016	15/06/2016		victor
36		Supervisión y corrección de la documentación	2d?	16/06/2016	17/06/2016	35	victor
37		Preparación de la lectura	5d?	20/06/2016	24/06/2016	36	victor

**Figura 46 : Tareas y dedicación Sprint 4 final**

### 18.1.6 Diagrama de Gantt final

A continuación mostraremos el diagrama de Gantt del proyecto final. Lo subdividiremos en 2 partes para una mejor visión del mismo.

#### 18.1.6.1 Parte Diagrama de Gantt final: Sprint Inception

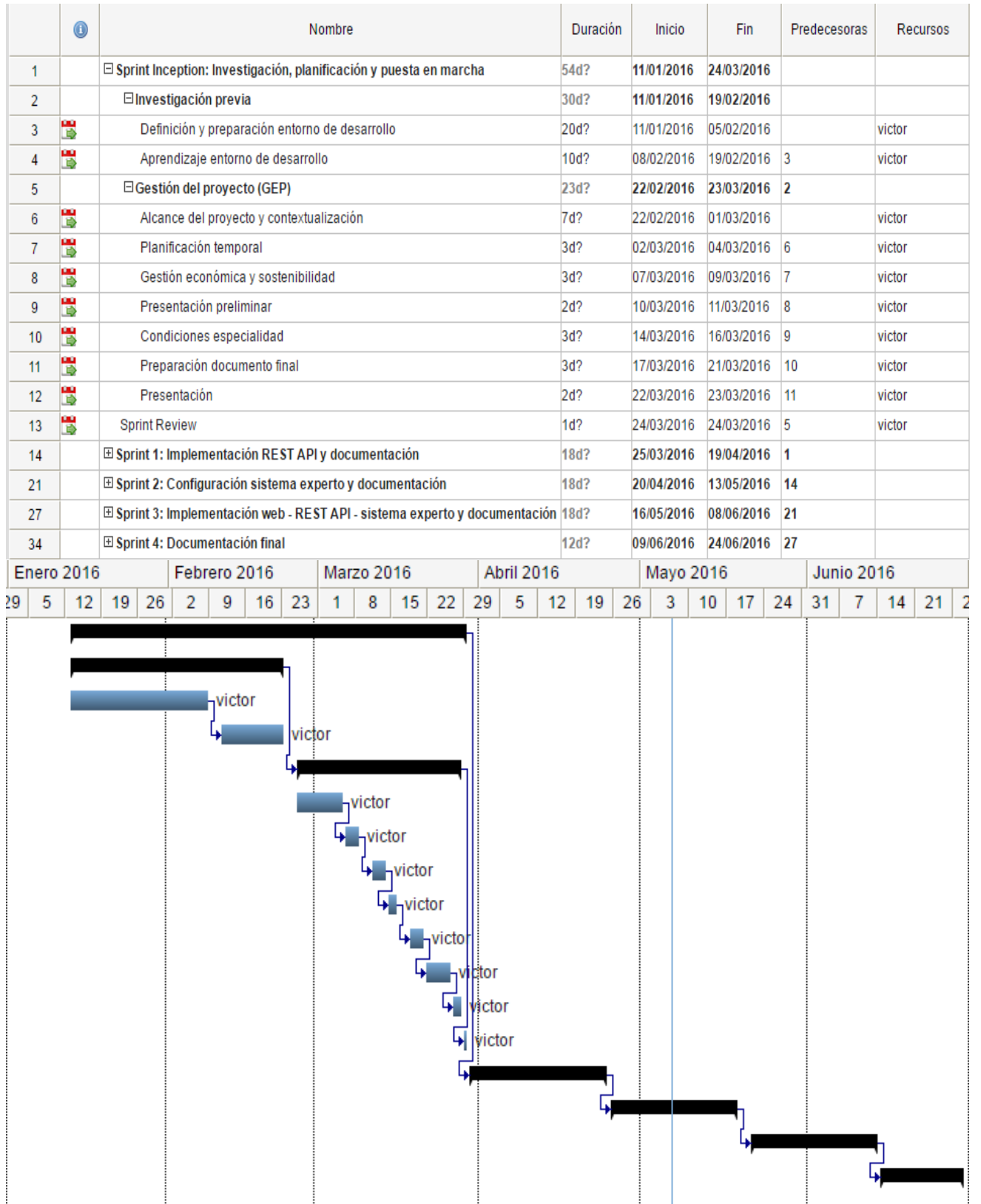


Figura 47: Diagrama de Gantt Sprint Inception final

### 18.1.6.2 Parte Diagrama de Gantt final: Sprint 1, 2, 3, 4

	①	Nombre	Duración	Inicio	Fin	Predecesoras	Recursos
1		☐ Sprint Inception: Investigación, planificación y puesta en marcha	54d?	11/01/2016	24/03/2016		
14		☐ Sprint 1: Implementación REST API y documentación	18d?	25/03/2016	19/04/2016	1	
15		Definición estructura REST API	3d?	25/03/2016	29/03/2016		victor
16		Implementación método POST	5d?	30/03/2016	05/04/2016	15	victor
17		Implementación método GET	5d?	06/04/2016	12/04/2016	16	victor
18		Testing	2d?	13/04/2016	14/04/2016	17	victor
19		Redacción memoria final	2d?	15/04/2016	18/04/2016	18	victor
20		Sprint Review	1d?	19/04/2016	19/04/2016	19	victor
21		☐ Sprint 2: Configuración sistema experto y documentación	18d?	20/04/2016	13/05/2016	14	
22		Definición sistema experto	3d?	20/04/2016	22/04/2016		victor
23		Configuración sistema experto	10d?	25/04/2016	06/05/2016	22	victor
24		Testing	2d?	09/05/2016	10/05/2016	23	victor
25		Redacción memoria final	2d?	11/05/2016	12/05/2016	24	victor
26		Sprint Review	1d?	13/05/2016	13/05/2016	25	victor
27		☐ Sprint 3: Implementación web - REST API - sistema experto y documentación	18d?	16/05/2016	08/06/2016	21	
28		Definición estructura	3d?	16/05/2016	18/05/2016		victor
29		Implementación web	5d?	19/05/2016	25/05/2016	28	victor
30		implementación web - REST API - sistema experto	5d?	26/05/2016	01/06/2016	29	victor
31		Testing	2d?	02/06/2016	03/06/2016	30	victor
32		Documentación memoria final	2d?	06/06/2016	07/06/2016	31	victor
33		Sprint Review	1d?	08/06/2016	08/06/2016	32	victor
34		☐ Sprint 4: Documentación final	12d?	09/06/2016	24/06/2016	27	
35		Redacción de la memoria	5d?	09/06/2016	15/06/2016		victor
36		Supervisión y corrección de la documentación	2d?	16/06/2016	17/06/2016	35	victor
37		Preparación de la lectura	5d?	20/06/2016	24/06/2016	36	victor

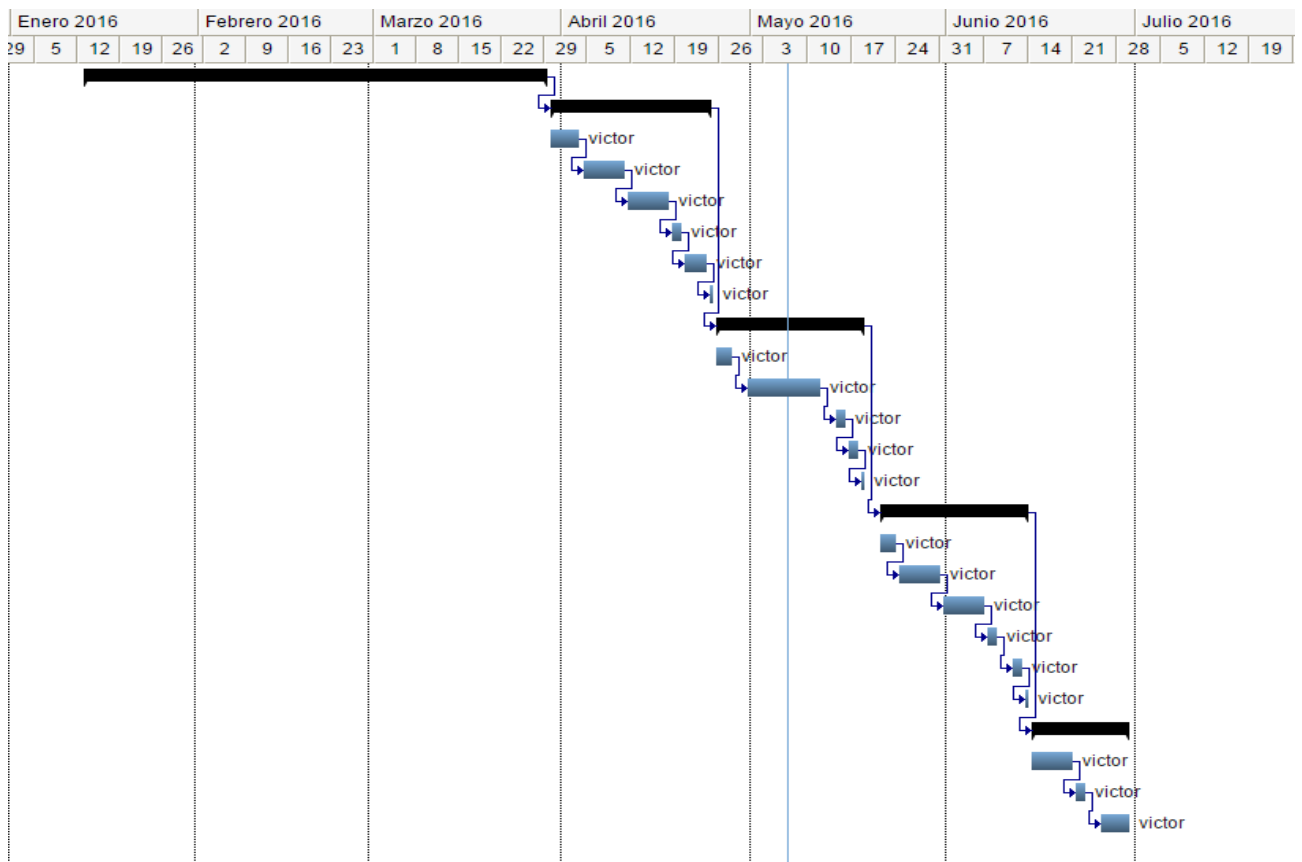


Figura 48: Diagrama de Gantt Sprint 1,2,3,4 final



## 18.2 Análisis económico final

Tras haber anotado todas las modificaciones que ha habido en la planificación del mismo respecto a la planificación inicial que se realizó del proyecto, se ha estimado el coste económico final del proyecto.

### 18.2.1 Costes directos por actividad

Por lo que respecta a los costes directos que hacen referencia a las remuneraciones de los recursos humanos, no ha habido ninguna variación respecto al análisis económico inicial. Siendo estos los siguientes:

Rol	Remuneración anual (€)	Remuneración (€/h)
Jefe de proyecto	35000 €	18,23 €/h
Analista funcional	33000 €	17,19 €/h
Diseñador	18000 €	9,38 €/h
Programador	18000 €	9,38 €/h
Responsable de pruebas	22000 €	11,46 €/h

*Tabla 15: Remuneraciones salariales según el rol*

En la tabla siguiente mostramos los costes estimados de cada actividad del proyecto definitivos.

Actividad	Unidades	Rol	Precio unitario(€/h)	Coste total (€)
Definición	40 h	Analista funcional	17,19 €/h	687,6 €
Preparación y aprendizaje entorno de desarrollo	80 h	Programador	9,38 €/h	750,4 €
Gestión del proyecto (GEP)	92 h	Jefe de proyecto	18,23 €/h	1677,16 €
Sprint Review	4 h	Jefe de proyecto	18,23 €/h	72,92 €
Sprint 1: Definición e implementación	65 h	Programador	9,38 €/h	609,7 €
Sprint 1: Testing	10 h	Responsable de pruebas	11,46 €/h	114,6 €
Sprint 1: Redacción memoria final	10 h	Jefe de proyecto	18,23 €/h	182,3 €
Sprint 1: Review	5 h	Jefe de proyecto	18,23 €/h	91,15 €
Sprint 2:	65 h	Programador	9,38 €/h	609,7 €

Definición y configuración				
<b>Sprint 2: Testing</b>	10 h	Responsable de pruebas	11,46 €/h	114,6 €
<b>Sprint 2: Redacción memoria final</b>	10 h	Jefe de proyecto	18,23 €/h	182,3 €
<b>Sprint 2: Review</b>	5 h	Jefe de proyecto	18,23 €/h	91,15 €
<b>Sprint 3: Definición e implementación</b>	65 h	Programador	9,38 €/h	609,7 €
<b>Sprint 3: Testing</b>	10 h	Responsable de pruebas	11,46 €/h	114,6 €
<b>Sprint 3: Redacción memoria final</b>	10 h	Jefe de proyecto	18,23 €/h	182,3 €
<b>Sprint 3: Review</b>	5 h	Jefe de proyecto	18,23 €/h	91,15 €
<b>Sprint 4: Documentación final</b>	48 h	Jefe de proyecto	18,23 €/h	875,04 €
<b>Total</b>	<b>534 h</b>			<b>7056,37€</b>

*Tabla 16: Costes directos por actividad finales*

También como coste directo se tiene en cuenta el hardware y software empleados en el proyecto, pero en este caso no ha habido ninguna variedad respecto al análisis inicial.

Producto	Unidades	Precio unitario	Dedicación	Coste estimado
<b>Amortización de la maquinaria</b>	5 meses	750 €/4 años	80%	62,5 €
<b>Amortización del software</b>	5 meses	31200 €/año	100%	13000 €
<b>Total</b>				<b>13062,5 €</b>

*Tabla 17: Costes directos maquinaria/software finales*

En el análisis inicial, se estimó que se dedicarían 480 horas mientras que en el análisis final nos ha salido de 534 horas, lo que ha provocado una desviación de 54 horas y por lo tanto en un aumento de 598.65 € sobre lo inicialmente previsto en los costes directos del proyecto.

El coste directo total empleado es de 20118,87 €

	Coste estimado final
<b>Roles</b>	7056,37€ €
<b>Maquinaria y software</b>	13062,5 €
<b>Total</b>	<b>20118,87 €</b>

*Tabla 18: Costes directo total final*

### 18.2.2 Costes indirectos

Los costes indirectos inicialmente previstos en el análisis económico inicial se han mantenido sin ninguna variación.

Producto	Unidades	Precio unitario	Dedicación	Coste estimado
<b>Conexión de datos</b>	5 meses	36,18 €/mes	50%	90,45 €
<b>Transporte</b>	5 meses	42,50 €/mes	100%	212,5 €
<b>Impresión en papel</b>	1200 paginas	0,05 €/pagina	100%	60 €
<b>Total</b>				<b>362,95 €</b>

*Tabla 19: Costes indirecto total final*

### 18.2.3 Presupuesto final

En el presupuesto final del proyecto tenemos en cuenta costes directos e indirectos, los cuales dan un resultado final de 20481,82 €.

Concepto	Coste de mercado
<b>Costes directos</b>	20118,87 €
<b>Costes Indirectos</b>	362,95 €
<b>Total</b>	<b>20481,82 €</b>

*Tabla 20: Presupuesto final del proyecto*

Teniendo en cuenta que en el análisis económico inicial se tuvo en cuenta costes por contingencia e imprevistos, el presupuesto final del proyecto no ha sufrido grandes desviaciones. En el análisis inicial se tuvo en cuenta por ejemplo costes por posibles averías de la maquinaria utilizada, los cuales no han sucedido a lo largo del proyecto. Como el presupuesto inicial marcado fue de 24009,85 € como refleja la siguiente tabla.

Concepto	Coste de mercado
Costes directos	19520,22 €
Costes Indirectos	362,95 €
Contingencia	3976,64 €
Imprevistos	150,04 €
<b>Total</b>	<b>24009,85 €</b>

*Tabla 21: Costes y presupuestos iniciales*

Hemos conseguido un ahorro (desviación negativa) de 3528,03 €, con un presupuesto final de 20481,82 €.

Concepto	Coste de mercado
Costes económico inicial	24009,85 €
Costes económico final	20481,82 €
<b>Total ahorro</b>	<b>3528,03 €</b>

*Tabla 22: Diferencia coste presupuesto final-inicial*

### 18.3 Metodología

La metodología inicialmente explicada que se basaba en metodologías ágiles, no ha sufrido ningún cambio a lo largo del proyecto y se ha ido siguiendo la pauta marcada por esta de un seguimiento constante con reuniones y entregas periódicas con tal de comprobar y determinar que los objetivos del proyecto se iban cumpliendo sobre lo inicialmente previsto. Esto ha hecho que a cada entrega y reunión se fuera perfeccionando cada vez más las previsiones de entregas futuras.

## **19 Integración de conocimientos**

A la hora de diseñar este proyecto, se han aprendido diferentes tecnologías nuevas, así como la aplicación de conocimientos adquiridos a lo largo de toda la carrera. Comenzando por la programación fundamental adquirida a lo largo de la misma con asignaturas como PRO1, PRO2, EDA, PROP, o aquellas específicas de la especialidad de Ingeniería del Software como BD, IES, AS, ASW, GPS, ER, PES, DBD, CBDE, CSI. También asignaturas de otras especialidades como APC.

Para la realización de este proyecto, las siguientes asignaturas realizadas son las que me han aportado mayor ayuda con sus conocimientos:

### **19.1 ASW (Aplicaciones y Servicios Web)**

Esta asignatura nos ha aportado conocimientos sobre programación y diseño de aplicaciones y servicios web, donde se ha tocado JavaScript, HTML, CSS, JSPs, JavaServlets y XML que son utilizados en este proyecto.

### **19.2 DBD y BD (Diseño de Base de Datos y Base de Datos)**

Estas asignaturas nos han aportado conocimientos sobre utilización de bases de datos.

### **19.3 PES (Proyecto de Ingeniería del Software)**

Esta asignatura nos ha aportado conocimientos sobre todos los factores y elementos que se han de tener en cuenta en la realización de un proyecto de software, como pueden ser la planificación del mismo, costes, diseño de requisitos, casos de uso o arquitectura, programación y presentaciones orales.

### **19.4 PROP (Proyectos de Programación)**

Esta asignatura nos ha aportado conocimientos sobre programación orientada a objetos en Java, así como diseño, programación, especificación de casos de uso y diagramas de clases.

### **19.5 APC y CSI (Arquitectura del PC y Conceptos de sistemas de información)**

Estas asignaturas nos han aportado conocimientos sobre cómo realizar y preparar presentaciones orales públicas. Desde cómo hacer las diapositivas, a que gestualización, tono de voz o puesta en escena hay que utilizar en cada momento.

## 20 Listado de competencias técnicas del proyecto

Las diferentes competencias técnicas que se tratan en este proyecto son las siguientes:

- **CES1.1:** Desarrollar, mantener y evaluar sistemas y servicios software complejos y/o críticos. [En profundidad]

Este proyecto debe presentar gran estabilidad y tener la capacidad de soportar gran cantidad de usuarios e información. Por todo ello los posibles estados críticos son controlados con tal de que no haya errores previamente.

- **CES1.6:** Administrar bases de datos (CIS4.3) [Bastante]

El sistema trabajara con una base de datos, que se administra y gestiona correctamente para su correcto funcionamiento.

- **CES1.7:** Controlar la calidad y diseñar pruebas en la producción de software. [En profundidad].

Para garantizar que el producto final no contiene ningún tipo de error, se han realizado pruebas al sistema al final de cada Sprint para asegurarse del correcto funcionamiento e integración de este.

- **CES2.1:** Definir y gestionar los requisitos de un sistema software. [En profundidad].

Este proyecto va destinado a cualquier tipo de empresa, en este caso concretamente a la empresa Everis, por todo ello el sistema ha de satisfacer los requisitos. El sistema ha de pasar obligatoriamente una fase de análisis de requisitos con tal de asegurarnos que el software a desarrollar soluciona la problemática a los usuarios finales.

- **CES2.2:** Diseñar soluciones apropiadas en uno o más dominios de aplicación, utilizando métodos de ingeniería del software que integren aspectos éticos, sociales, legales y económicos. [Bastante]

Este proyecto está dentro de unos parámetros económicos aptos para las empresas, así como no perjudicar a ningún tipo de grupo social, y aportar una mejora social ya que las empresas podrán ofrecer a sus usuarios o clientes aquello que soliciten de una manera inmediata.

## **21 Leyes y Regulaciones**

En este proyecto no se trata ni se involucra datos sensibles de ningún tipo, y por tanto no interviene ningún tipo de ley o regulación sobre el mismo.

## 22 Mejoras

En este apartado, trataremos posibles mejoras en el sistema que se podrían aplicar en un futuro.

Una de las partes fundamentales y claves en este sistema son los datos que obtenemos de la interacción de los usuarios por el portal web. En este aspecto nosotros hemos tratado con ciertos datos que nos interesaban como por ejemplo el identificador de sesión o la ruta por la cual navega, pero se podrían tratar muchos más en función de la información de la web y las solicitudes o necesidades de las empresas. Cuanta más información se trate más efectivo será el sistema.

Otro aspecto que se podría mejorar en un futuro es el formato de visualización de datos en la Rest API, pudiéndolo mostrar en Json o txt según el usuario lo desee y no únicamente XML.

Por último sería mejorar el estilo de los portales web, con tal de hacer la apariencia visual más efectiva y agradable para los usuarios, así como aumentar las funcionalidades de esta.



## 23 Situación actual y futuro

Tras la finalización de este proyecto, disponemos de un prototipo básico de sistema que es capaz de obtener, estructurar, distribuir y analizar los datos de la interacción de los usuarios por los portales web. Este sistema es adaptable y escalable para cualquier tipo de empresa que disponga de un portal web, ofreciéndole una mejora en el análisis de datos.

Aun así este proyecto sigue siendo un prototipo inicial, donde se han implementado las funcionalidades más importantes y básicas de cara a tener una estructura inicial para que en el futuro pueda ser mejorada y se pueda considerar el sistema apto de cara a su comercialización.

Este proyecto podría ser implantado en cualquier proyecto de la empresa para la cual realizo el mismo, Everis, con lo cual el conocimiento adquirido será fundamental y de gran ayuda a la hora de aplicarlo.

## 24 Conclusiones

Después de la finalización de este proyecto, se han conseguido alcanzar los objetivos principales del mismo que fueron definidos al inicio y que son la creación de una Rest API capaz de obtener, estructurar y distribuir los datos de la interacción de los usuarios por el portal web, así como la creación de un portal web, y por último la configuración del sistema experto que nos permite hacer el análisis e interpretación de estos datos.

A pesar de los imprevistos surgidos y las modificaciones que se han tenido que realizar respecto a la planificación inicial realizada, estoy muy satisfecho con la realización del trabajo de este proyecto. Gracias a la elaboración del proyecto he aprendido tecnologías nuevas que no había utilizado previamente ya que las desconocía, así como profundizar y adquirir muchos más conocimientos sobre tecnologías o lenguajes en los que tenía una base de conocimiento ya que los había utilizado previamente. Además he podido ver el alcance de un proyecto, desde la definición y planificación inicial hasta la entrega final.

También el hacer el proyecto para una empresa, en este caso Everis, me ha permitido ver como es el funcionamiento día a día del desarrollo de un proyecto, ya que he seguido los métodos que utiliza la empresa para hacer el seguimiento de estos.

Por todo esto, este proyecto me ha servido para adquirir muchos conocimientos que me serán de gran utilidad en mi presente y futuro laboral.

## 25 Referencias

[1]Microsoft Azure ML [Consultado: 23-02-2016].

<https://azure.microsoft.com/es-es/services/machine-learning/>

[2]Google Prediction Api [Consultado: 23-02-2016].

<https://cloud.google.com/prediction/>

[3]Big ML [Consultado: 23-02-2016].

<https://bigml.com/>

[4]PredicSis [Consultado: 24-02-2016].

<http://www.predicsis.com/>

[5]IBM Watson Analytics [Consultado: 24-02-2016].

<https://www.ibm.com/marketplace/cloud/watson-analytics/us/en-us>

[6]Liferay [Consultado: 10-02-2016].

<https://www.liferay.com/es/>

[7]Amazon [Consultado: 25-02-2016].

<https://aws.amazon.com/es/machine-learning/>

<https://aws.amazon.com/>

[8]Toad MySQL [Consultado: 15-02-2016].

<https://www.toadworld.com/>

[9]MySQL Workbench [Consultado: 16-02-2016].

<https://www.mysql.com/products/workbench/>

## 26 Glosario

**API:** La API es una interfaz de programación de aplicaciones que especifica como componentes informáticos diferentes pueden interactuar.

**REST API:** La REST API es un tipo de arquitectura de software para sistemas web que nos permite describir interfaces entre sistemas mediante HTTP, por lo tanto se accede a través de URLs o direcciones web. Los datos que se obtienen de estas consultas pueden ser en diferentes formatos como pueden ser JSON o XML entre otros.

**JSON:** Estándar de texto que sirve para representar estructuras de datos simples para el intercambio de datos entre aplicaciones.

**XML:** Lenguaje de marcas utilizado para el almacenamiento de datos de una forma legible.

**HTML:** Lenguaje de marcas de hipertexto para la elaboración de páginas web.

**CSS:** Lenguaje que define y crea la presentación de documentos estructurados como HTML o XML.

**Java:** Lenguaje de programación orientado a objetos.

**Liferay:** Portal de gestión de contenidos de código abierto escrito en Java.

**Eclipse:** Plataforma de software compuesta por un conjunto de herramientas con el fin de desarrollar aplicaciones basadas en navegadores.

**SaaS:** Modelo de distribución de software, donde los datos y el soporte lógico son gestionados por una empresa externa y a la que el cliente puede acceder a ella a través de internet.

**OAuth:** Protocolo de autorización para aplicaciones informáticas o lugares web.